

**IDC DOCUMENTATION**

**Station  
Processing  
(StaPro)  
Software  
User Manual**



**Notice**

This document was published September 2001 by the Monitoring Systems Operation of Science Applications International Corporation (SAIC) as part of the International Data Centre (IDC) Documentation. Every effort was made to ensure that the information in this document was accurate at the time of publication. However, information is subject to change.

**Contributors**

Richard D. Jenkins, Science Applications International Corporation

**Trademarks**

BEA TUXEDO is a registered trademark of BEA Systems, Inc.  
Ethernet is a registered trademark of Xerox Corporation.  
NeuralWare Professional II by MathWorks.  
ORACLE is a registered trademark of Oracle Corporation.  
PL/SQL is a trademark of Oracle Corporation.  
SAIC is a trademark of Science Applications International Corporation.  
Solaris is a registered trademark of Sun Microsystems.  
SQL\*Plus is a registered trademark of Oracle Corporation.  
Sun is a registered trademark of Sun Microsystems.  
UltraSPARC is a registered trademark of Sun Microsystems.  
UNIX is a registered trademark of UNIX System Labs, Inc.

**Ordering Information**

The ordering number for this document is SAIC-01/3036.

This document is cited within other IDC documents as IDC-6.5.11.

# Station Processing (StaPro) Software User Manual

## CONTENTS

<b>About this Document</b>	i
■ PURPOSE	ii
■ SCOPE	ii
■ AUDIENCE	ii
■ RELATED INFORMATION	ii
■ USING THIS DOCUMENT	iii
Conventions	iv
<b>Chapter 1: Introduction</b>	1
■ SOFTWARE OVERVIEW	2
■ STATUS OF DEVELOPMENT	4
■ FUNCTIONALITY	5
Features and Capabilities	5
Performance Characteristics	6
Related Tools	6
■ INVENTORY	7
■ ENVIRONMENT AND STATES OF OPERATION	11
Software Environment	11
Normal Operational State	11
<b>Chapter 2: Operational Procedures</b>	13
■ SOFTWARE STARTUP	14
Automatic	14
Manual	14
■ BASIC PROCEDURES	15
Setting Parameters	16
■ ADVANCED PROCEDURES	26

Using Neural Networks	26
Using CLIPS Rules	37
Tuning Stations	42
■ MAINTENANCE	44
Maintaining Log Files	45
■ SECURITY	45
Passwords	45
<b>Chapter 3: Troubleshooting</b>	47
■ MONITORING	48
Processing-status Display	48
Database Queries	48
Log Monitoring	49
■ INTERPRETING ERROR MESSAGES	49
Error Exit Messages	49
Non-exit Messages	52
■ SOLVING COMMON PROBLEMS	54
Error Recovery	55
<b>Chapter 4: Installation Procedures</b>	57
■ PREPARATION	58
Obtaining Released Software	58
Hardware Mapping	58
UNIX System	58
■ EXECUTABLE FILES	59
■ CONFIGURATION DATA FILES	59
■ DATABASE	59
Accounts	59
Tables	60
Initialization of lastid	60
■ TUXEDO FILES	60
■ INITIATING OPERATIONS	60
■ VALIDATING INSTALLATION	61

<b>References</b>	63
<b>Glossary</b>	G1
<b>Index</b>	I1



# Station Processing (StaPro) Software User Manual

## FIGURES

FIGURE 1.	IDC SOFTWARE CONFIGURATION HIERARCHY	3
FIGURE 2.	RELATIONSHIP OF STAPRO TO OTHER SOFTWARE UNITS OF AUTOMATIC PROCESSING CSCI	4





# Station Processing (StaPro) Software User Manual

## TABLES

TABLE I:	DATA FLOW SYMBOLS	iv
TABLE II:	TYPOGRAPHICAL CONVENTIONS	v
TABLE 1:	DATABASE TABLES USED BY STAPRO	7
TABLE 2:	FLAT FILES USED BY STAPRO	8
TABLE 3:	INPUTS TO SEISMIC NEURAL NETWORK	27
TABLE 4:	FORMAT OF SEISMIC NEURAL NETWORK WEIGHTS FILE	30
TABLE 5:	INPUTS TO HYDROACOUSTIC NEURAL NETWORK	32
TABLE 6:	FORMAT OF HYDROACOUSTIC NEURAL NETWORK WEIGHTS FILE	35
TABLE 7:	STAPRO FUNCTIONS BOUND TO CLIPS	37
TABLE 8:	CONTROL FACTS AVAILABLE TO CLIPS RULES	39
TABLE 9:	GENERAL CHECKLIST FOR ADDING NEW STATION	42
TABLE 10:	STAPRO-SPECIFIC CHECKLIST FOR ADDING NEW STATION	43



## About this Document

This chapter describes the organization and content of the document and includes the following topics:

- Purpose
- Scope
- Audience
- Related Information
- Using this Document

## About this Document

## PURPOSE

This document describes how to use the Station Processing (*StaPro*) software of the International Data Centre (IDC). The software is a computer software component (CSC) of the Automatic Processing Computer Software Configuration Item (CSCI) and is identified as follows:

Title: Station Processing

Abbreviation: *StaPro*

## SCOPE

This manual includes instructions for setting up the software, using its features, and basic troubleshooting. This document does not describe the software's design. This topic is described in sources cited in "Related Information."

## AUDIENCE

This document is intended for the first-time or occasional user of the software. However, more experienced users may find certain sections useful as a reference.

## RELATED INFORMATION

The following documents complement this document:

- *Station Processing (StaPro) Software* [IDC7.1.12]
- *Database Schema* [IDC5.1.1Rev2]
- *Configuration of PIDC Databases* [IDC5.1.3Rev0.1]

- *IDC Processing of Seismic, Hydroacoustic, and Infrasonic Data* [IDC5.2.1]

See “References” on page 63 for a list of documents that supplement this document. The *StaPro* UNIX manual (man) page applies to the existing *StaPro* software.

## USING THIS DOCUMENT

This document is part of the overall documentation architecture for the IDC. It is part of the Technical Instructions category, which provides guidance for installing, operating, and maintaining the IDC systems. This document is organized as follows:

- Chapter 1: Introduction  
This chapter provides an overview of the software’s capabilities, development, and operating environment.
- Chapter 2: Operational Procedures  
This chapter describes how to use the software and includes detailed procedures for startup and shutdown, basic and advanced features, security, and maintenance.
- Chapter 3: Troubleshooting  
This chapter describes how to identify and correct common problems related to the software.
- Chapter 4: Installation Procedures  
This chapter describes first how to prepare for installing the software, then how to install the executable files, configuration data files, database elements, and Tuxedo files. It also describes how to initiate operation and how to validate the installation.
- References  
This section lists the sources cited in this document.
- Glossary  
This section defines the terms, abbreviations, and acronyms used in this document.

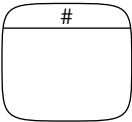

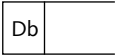
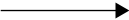
▼ About this Document

- Index
- This section lists topics and features provided in this document along with page numbers for reference.

Conventions

This document uses a variety of conventions, which are described in the following tables. Table I shows the conventions for data flow diagrams. Table II lists typographical conventions.

TABLE I: DATA FLOW SYMBOLS

Description	Symbol <sup>1</sup>
process	
external source or sink of data	
data store Db = database store	
data flow	

1. Symbols in this table are based on Gane-Sarson conventions [Gan79].

TABLE II: TYPOGRAPHICAL CONVENTIONS

Element	Font	Example
database table	<b>bold</b>	<b>detection</b>
database table and attribute when written in the dot notation		<b>hydro_features.low_cut</b>
database attributes	<i>italics</i>	<i>peak_level</i>
processes, software units, and libraries		<i>StaPro</i>
user-defined arguments and variables used in parameter (par) files or program command lines		StaPro par= <i>my.par</i>
titles of documents		<i>Database Schema</i>
computer code and output	<b>courier</b>	max-delslo=999.9
filenames, directories, and websites		tuxshell-StaPro.par
text that should be typed exactly as shown		StaPro sta=ARCES





# Chapter 1: Introduction

This chapter provides a general description of the software and includes the following topics:

- Software Overview
- Status of Development
- Functionality
- Inventory
- Environment and States of Operation

# Chapter 1: Introduction

## SOFTWARE OVERVIEW

The purpose of the *StaPro* software is to classify detections as arrivals with various phase types. Figure 1 shows the logical organization of the IDC software. *StaPro* is a component of the Station Processing CSC of the Automatic Processing CSCI.

Figure 2 shows the relationship of *StaPro* to other components of the Automatic Processing CSCI. The Continuous Data Subsystem receives data from primary seismic, hydroacoustic, and infrasonic (S/H/I) stations and stores this information in the station processing (Db1 in Figure 2) database. The data consist of ancillary information stored in ORACLE database tables and binary waveform files on the UNIX filesystem. The Detection and Feature Extraction program (*DFX*) reads the waveform data and applies a variety of signal processing algorithms to make detections and extract feature measurements such as: onset time, amplitude, period, and rectilinearity, for each of the detections. *DFX* writes these features to tables in the station processing database. The *tuxshell* software of the Distributed Processing CSCI monitors the *StaPro* queue and schedules *StaPro* to run after *DFX* has completed. *StaPro* reads parametric feature measurements from the database, determines the most likely phase names (for example, Pn, Pg, Sn, Lg), and groups related arrivals. Estimates of single-station location and magnitude are computed for seismic stations. Results are stored in the station processing database. The Global Association (GA) Subsystem uses the phase identifications (IDs), groupings, and magnitude from *StaPro* to associate signals recorded at multiple stations and define events. GA reads *StaPro* results from the station processing database and writes its results to the network processing database (Db2 in Figure 2). *WaveExpert* selects auxiliary waveform data that are expected to improve the event location; and for these stations, time intervals are retrieved by the Retrieve Subsystem. Although not shown in Figure 2, *DFX* and *StaPro* are also executed for auxiliary data.

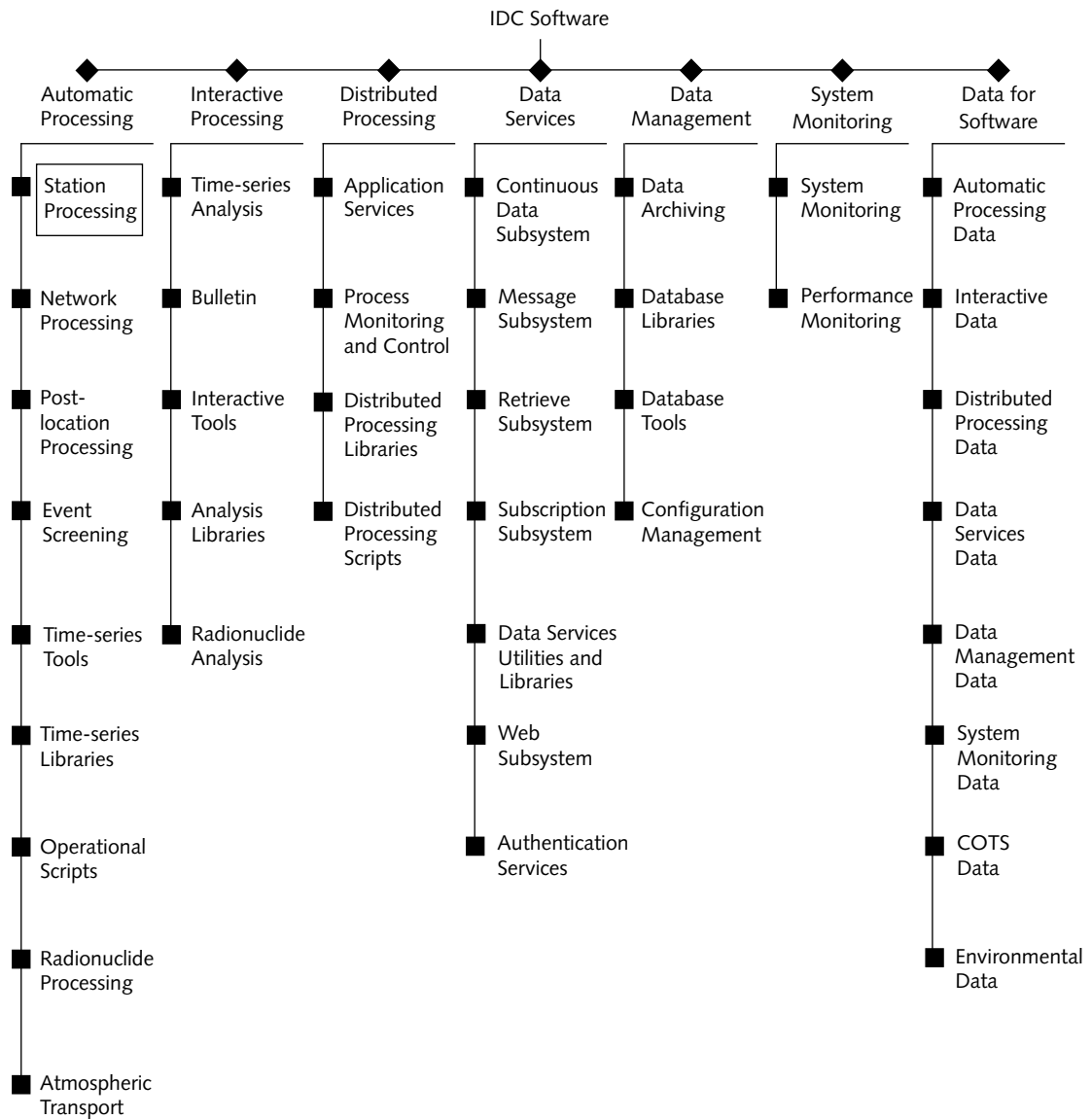


FIGURE 1. IDC SOFTWARE CONFIGURATION HIERARCHY

## ▼ Introduction

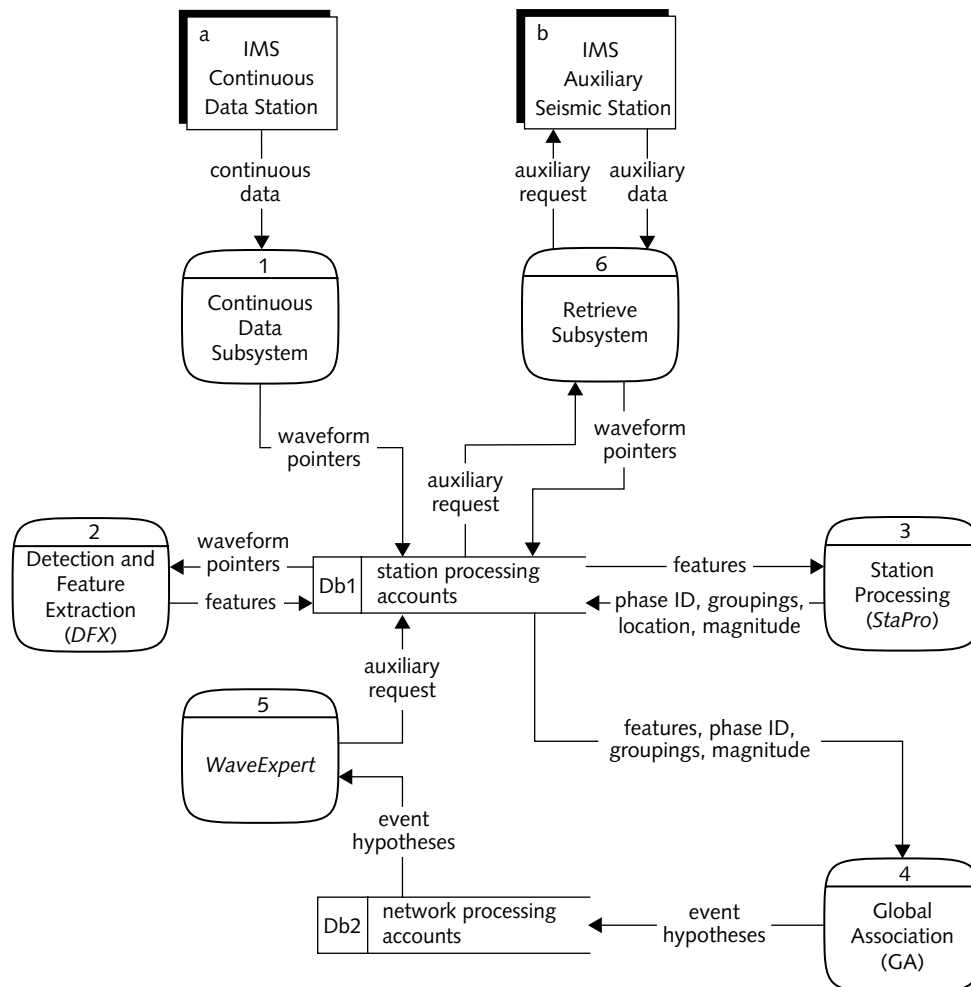


FIGURE 2. RELATIONSHIP OF STAPro TO OTHER SOFTWARE UNITS OF AUTOMATIC PROCESSING CSCI

## STATUS OF DEVELOPMENT

This document describes mature software that has been used operationally since 1994 at the Prototype International Data Centre (PIDC) and since 1998 at the IDC in Vienna, Austria. The software is based on algorithms used in the *ESAL* applica-

tion program, which was created for the early Group of Scientific Experts Technical Test (GSETT) experiments. Since its inception, *StaPro* has been extended to include hydroacoustic station processing and infrasonic station processing.

## FUNCTIONALITY

The *StaPro* software is organized into modular functions that together form the code base for single-station processing. The core functionality of *StaPro* consists of three functions: (1) determining signal type, (2) grouping signals, and (3) identifying phases. Two additional functions are available for processing seismic data. These functions include code for estimating single station location and local magnitude  $M_L$ .

### Features and Capabilities

This section highlights several important features and capabilities that explain *StaPro's* functionality. The topics introduced in this section include processing multi-technology data, use of a Generic Database Interface (GDI), use of a neural network, application of slowness and azimuth station corrections, convenience of using station-specific parameters, and the ability to estimate hypocenter locations and magnitudes.

An important capability of *StaPro* is that it is designed to process data from three technologies: (1) seismic, (2) hydroacoustic, and (3) infrasonic. The seismic processing code is the most mature part of *StaPro*, and infrasonic processing is the least mature.

Feature measurements of detections and *StaPro* results are stored in the database. *StaPro* uses the GDI functionality to transfer information between *StaPro* and the database.

A neural network, an advanced feature available in *StaPro*, helps determine the signal type of detections. This feature requires off-line calculation of neural-network weights that define an empirically determined non-linear boundary between signal types and noise. For more information see "Advanced Procedures" on page 26.

## ▼ Introduction

Slowness and azimuth station corrections (SASCs) can be applied to individual stations in *StaPro*. The corrections are read during the initialization step and are applied to slowness and azimuth values stored in an internal data structure. The corrected values are used internally in *StaPro* but are not updated in the database.

User flexibility is a design feature of *StaPro*. The software can be easily configured for individual stations. Station-specific parameters may be set or default values used.

Estimates of epicentral location and event size are features that are only calculated when processing seismic data. At least one P phase and one S phase are required to form an estimate.

**Performance Characteristics**

*StaPro* runs relatively fast compared to *DFX* for the same station and time segment. The processing time required by *StaPro* to complete a 10-minute segment of seismic data is approximately 2 minutes on a SPARC 20 Model 612. This time varies (1–3 minutes) depending on the number of detections to be processed and the hardware configuration.

*StaPro* is very reliable and rarely exits with an error. In some cases, the accuracy of phase names, and hence detection groups determined by *StaPro*, can be improved. The complexity of some detections makes it difficult to resolve phase names using the available information. Improvements in *DFX* feature measurements and/or inclusion of new feature characteristics may help improve *StaPro* accuracy.

**Related Tools**

*StaPro* is a self-contained process. A related tool called *nnet\_train* is available for training neural-network weights. This tool is used to train weights for the three-component (3-C) stations (see [Wan97a] and [Wan98c]).

## INVENTORY

This section identifies various items needed to run *StaPro*. The UNIX filesystem organization and database naming convention follow the PIDC model. This model uses the station processing database account to store results. Table 1 lists the database tables used by *StaPro*. Some are read-only tables while others are used to store processing results.

**TABLE 1: DATABASE TABLES USED BY STAPRO**

Name	Mode	Description
<b>amp3c</b>	R	This table contains horizontal-to-vertical amplitude ratios of detection found in 3-C station data.
<b>amplitude</b>	R	This table contains amplitudes and periods of detections.
<b>apma</b>	R	This table contains the results of particle motion analysis for detections found in 3-C station data.
<b>arrival</b>	R/W	This table contains features for each detection. <i>StaPro</i> modifies <i>stassid</i> , <i>iphase</i> , <i>auth</i> , and <i>lddate</i> .
<b>assoc</b>	R/W	This table contains information about signals associated to single-station events. <i>StaPro</i> may delete old associations and write new ones to this table. This <b>assoc</b> table resides in the station processing database account.
<b>detection</b>	R	This table contains the F-statistic and f-k quality for each detection.
<b>hydro_features</b>	R	This table contains feature measurements for hydroacoustic detections.
<b>origerr</b>	R/W	This table contains summaries of confidence bounds in origin estimations for single-station events. <i>StaPro</i> may delete old <b>origerr</b> rows and write new ones to this table. This <b>origerr</b> table resides in the station processing database account.

## ▼ Introduction

TABLE 1: DATABASE TABLES USED BY STAPRO (CONTINUED)

Name	Mode	Description
<b>origin</b>	R/W	This table contains event location and magnitude estimates for single-station events. <i>StaPro</i> may delete old <b>origin</b> rows and write new ones to this table. This <b>origin</b> table resides in the station processing database account.
<b>site</b>	R	This table contains the location information and station type for each station.
<b>stassoc</b>	R/W	This table contains summary information about groups of related signals. <i>StaPro</i> may delete old station association members belonging to an old group. It may also write new station associations to this table.

Table 2 describes several input data files that are used by *StaPro*.

TABLE 2: FLAT FILES USED BY STAPRO

Name	Mode	Description
<code>StaPro.par</code>	R	This file contains general station processing parameters.
<code>bayes.tbl</code>	R	This file contains Bayesian inference tables.
<code>ipnnwts.tbl</code>	R	This file contains seismic neural network weights.
<code>hydro_\$(STA).weights</code>	R	This file contains hydroacoustic neural network weights.
<code>GA_dist_depth_ranges</code>	R	This file contains distance and depth limits for phases.
velocity model specification file	R	This file contains the velocity model file path and prefix.
magnitude definition file	R	This file contains the magnitude definition file path and prefix.
slowness-azimuth station correction file	R	This file contains the slowness-azimuth station correction file path.
<code>\$(STA).par</code>	R	This file contains station-specific parameters.



TABLE 2: FLAT FILES USED BY STAPRO (CONTINUED)

Name	Mode	Description
<code>shared.par</code>	R	This file contains system-level parameters.
<i>StaPro</i> log file	W	This file contains messages about the current <i>StaPro</i> session.
<code>\$(STA)_nnet.log</code>	W	This file contains a summary of neural network results.
<code>StaPro_loc.err</code>	W	This file contains a summary of <i>libloc</i> results.

The GDI is a shared object file that is linked to the *StaPro* executable. This file must be accessible when *StaPro* opens a database connection.

A system-level par file called `process.par` contains database password information that must be available to *StaPro* when the GDI is called. This file is read protected; only authorized users can read it. A second system-level par file called `shared.par` defines important data directory names. Both of these system-level par files are located in the UNIX directory `$(CMS_CONFIG)/system_specs/`.

A *tuxshell* par file called `tuxshell-StaPro.par` is used to define the *StaPro* command line for automatic pipeline operation. The command line is built from a station name, time segment, and the main *StaPro* par file. The *tuxshell* par file is located in the UNIX directory `$(CMS_CONFIG)/app_config/distributed/tuxshell/detpro/`.

The *StaPro* software consists of one executable of the same name. Documentation describing changes made to the software and major release dates is available in a file called `changes.StaPro`. This file is located in the source code area adjacent to the `$(CMS_CONFIG)` directory and is chronologically organized.

Parameters that control *StaPro* are listed in the `StaPro.par` file. Default values are listed when applicable. This file is located in the UNIX directory `$(CMS_CONFIG)/app_config/automatic/StaPro/`. Some station-specific *StaPro* parameters are

## ▼ Introduction

also listed in station par files located in the UNIX directory `$(CMS_CONFIG)/station_specs/$(STA).par`. A station-specific par file is referenced from within `StaPro.par`.

*StaPro* may also use data files during execution. Some of these are used only if they are available; *StaPro* will run without them. The SASCs are optional. The SASC files are specified in the *StaPro* parameter *sasc-dir-prefix* by giving the UNIX directory location and file prefix (for example, `$(CMS_CONFIG)/earth_specs/SASC/sasc`). Available station correction files are listed in the UNIX directory as `sasc.$(sta)`, and *StaPro* reads these files and makes use of the available corrections.

Neural-network weight files are optional for *StaPro* but are used if available. The files can be found in the UNIX directory `$(CMS_CONFIG)/earth_specs/STAPRO`. The filename for the seismic weights file is `ipnnwts.tbl`, and the hydroacoustic weights files are called `hydro_$(STA).weights`. Currently, stations PSUR, WK30, and WK31 have hydroacoustic weights.

A file called `bayes.tbl` located in the UNIX directory `$(CMS_CONFIG)/earth_specs/STAPRO` is required for *StaPro* when identifying phases during regional seismic-station processing. A Bayesian inference technique makes use of empirically derived probabilities (contained in the file) to decide the phase name of either the earliest S or the S with the largest amplitude.

The file called `GA_dist_depth_ranges`, located in the UNIX directory `$(CMS_CONFIG)/earth_specs/GA`, is used by *StaPro* to constrain the seismic phase-distance relationship. This file is also used by GA to apply similar constraints.

*StaPro* uses Velocity Model Specification File (VMSF), Magnitude Definition File (MDF), and Transmission Loss Specification File (TLSF) information during the location and magnitude calculations. These data files are found under the UNIX directory `$(CMS_CONFIG)/earth_specs/`.

## ENVIRONMENT AND STATES OF OPERATION

### Software Environment

*StaPro* is designed to run on a Sun workstation using the Solaris 7 operating system. *StaPro* does not require much processing power and therefore is run on the same machine as *DFX*. Together, *DFX* and *StaPro* comprise signal processing that is run in the automatic processing pipeline. The typical hardware configuration for signal processing consists of a Sun UltraSPARC machine with multiple processors, 1.5 GB of memory, and a minimum of 10 GB of magnetic disk. The required disk space scales with the number of days that log files are allowed to accumulate. *StaPro* obtains database services via an Ethernet connection to a local area network (LAN). ORACLE 8.1.5 is the current database software used at the PIDC and IDC. The UNIX filesystem is required to store configuration and parameter files, the GDI shared object file, and the *StaPro* executable.

### Normal Operational State

*StaPro* is normally invoked by the *tuxshell* program. The *tuxshell* program uses Tuxedo queues and the *tis\_server* program to register when a time segment has been successfully processed by *DFX*. After it is queued, *tuxshell* starts *StaPro*. The *tuxshell* par file contains the necessary parameters to build a proper *StaPro* command line. After execution, *StaPro* returns either a 0 (successful execution), or a 1 (a fatal error occurred). The *tuxshell* program can retry failed processing intervals up to a user-defined number of tries.



## Chapter 2: Operational Procedures

This chapter provides instructions for using the software and includes the following topics:

- Software Startup
- Basic Procedures
- Advanced Procedures
- Maintenance
- Security

## Chapter 2: Operational Procedures

## SOFTWARE STARTUP

## Automatic

In a normal operational environment *StaPro* is run as part of the automatic processing pipeline. The *tuxshell* program is configured to start *StaPro* directly after *DFX* has completed. Further details on how this is accomplished are provided in [IDC6.5.2Rev0.1]. The command line used by the *tuxshell* program is defined in the `tuxshell-StaPro.par` file. The command line contains system parameters and the *IMSPAR* environment variable. *IMSPAR* points to a system-level par file called `process.par`, which defines the needed parameters. Other station parameters may be added, but the nominal command line is as follows:

```
$(RELBIN)/StaPro sta=$(name) start-time=$(time) \
end-time=$(endtime) \
par=$(AUTOMATIC-DIR)/StaPro/StaPro.par
```

# Manual

The software may also be run manually by typing the desired command line. A convenient syntax is as follows:

```
StaPro par=my.par
```

where all parameters are either defined in the file *my.par* or in nested pars. This form is useful when off-line processing is required such as to run a test. For example, you may test *StaPro* with a lower teleseismic velocity threshold of 9.4 km/s rather than using the default value of 11.0 km/s. You can construct the *my.par* file as follows:

```
par=StaPro.par
# special settings follow
min-teleseismic-velocity=9.4
database=test/test@Ora815
```

In this example a private database containing *StaPro* input copied from the operations database that *StaPro* typically uses is specified. The parameter *database* is required to set the private database name. This example uses the specified value for *min-teleseismic-velocity* and *database*, but all other parameter settings are the same values used during normal operations as defined in *StaPro.par*.

## BASIC PROCEDURES

Basic operational procedures for *StaPro* consist of setting parameters that control database usage and processing thresholds for S/H/I (Seismic, Hydroacoustic, and Infrasonic) processing. This section describes these parameters, the configuration files, and environment variables needed to run *StaPro*.

The *StaPro.par* file sets system and other controlling variables through global par files before setting the application-specific parameters. The environment variable called *IMSPAR*, typically defined in `$(CMS_CONFIG)/system_specs/env/global.env`, points to another par file that defines many system-level parameters used in the operational system. This par file is called *process.par*, and it defines various database account and password combinations, such as *EXPERTDB*. It also references a nested par file called *shared.par* that defines important directory paths used by many applications (including *StaPro*). Some important parameters defined in this file are *VMSF*, *LOGDIR*, and *STAPRO\_SITE\_PATH*. *StaPro* parameters can also be found in the station par files `$(CMS_CONFIG)/station_specs/$(STA).par`. Each station has specific parameters saved in separate par files and some *StaPro*-specific parameters prefixed by "StaPro-". The specific `$(STA).par` is called from within *StaPro.par* and the "StaPro-" parameters are read. For example, the par file for seismic array station ARCES defines its own values for *noise-fkqual-fstat* and *min-teleseismic-velocity* parameters. The files for the hydroacoustic station VIB and infrasonic station WRAI define station type (*StaPro-statype*) as *hy* and *infra*, respectively. Both of these stations also set the

## ▼ Operational Procedures

*StaPro-do-loc-mag* flag to false (0) because event locations are not estimated for these technologies. In addition, the infrasonic stations define a longer window for *StaPro-infra-max-grouping-time*. Typically, no specific parameters are defined for 3-C seismic stations in the `$(CMS_CONFIG)/station_specs/$(STA).par` files.

*GDIHOME* is an important environment variable that is set as the top source-tree directory and is used by the GDI to find important shared object files needed to access the ORACLE database.

The *StaPro* parameters are organized into categories covering general program control, database, seismic processing, hydroacoustic processing, and infrasonic processing. Refer to the *StaPro* man page for the latest information on parameters and their default values.

## Setting Parameters

### Control Parameters

General control parameters define the station (*sta*) and time window to be processed (*start-time*, *end-time*, *duration*). These items must be specified because they do not have default values. Typically, *tuxshell* uses the same values for these parameters in the *StaPro* command line that it uses for *DFX*.

Other parameters in this category have default values and are not required unless you want to change their default setting. For example, the technology being processed is determined by the station type parameter (*statype*). Not setting this parameter causes *StaPro* to take a default action, which assumes that seismic data are being processed. This parameter must be specified to process hydroacoustic or infrasonic data.

The standard beam channel (*standard-beam-chan*) is used to obtain measurements from the amplitude table for seismic processing. The default is *v2040*, which refers to a 2–4 Hz filtered vertical channel beam. A different beam channel may be specified if needed.



*CLIPS* rules can be specified by setting the full UNIX path name to the rule file using the parameter *sta-rule-file*. If this parameter is not set, then *CLIPS* rules will not be used.

The *do-loc-mag* flag controls whether or not event location and magnitude are estimated. The default setting is true (1). For hydroacoustic and infrasonic processing this parameter must be set to false (0).

To apply special corrections for slowness and azimuth at seismic and infrasonic stations, set the *sasc-dir-prefix* parameter to the appropriate UNIX path and file prefix. The default action when this parameter is not set is that no corrections will be applied.

A flag (*verbose*) controls the volume of messages sent to the standard error device; it has a default value of 1. Higher values generate more messages. Values ranging from 0 to 9 are recognized.

### Database Connection Parameters

To connect to the ORACLE database, the name and password must be specified in the *database* parameter. The following seven parameters are the names of tables where input feature measurements are obtained: *amp3c-table*, *apma-table*, *amplitude-table*, *arrival-aux-table* (default is *detection*), *arrival-table*, *site-table*, and *hydro-table* (default is *hydro\_features*). Default table names are similar to the parameter name unless otherwise specified. The **arrival** and **amplitude** tables contain the bulk of the basic information needed by *StaPro*, such as the station name, onset time, amplitude, period, azimuth, and slowness. f-k quality and F-statistic estimates are obtained from **detection**. Additional information, such as horizontal-to-vertical power ratio and rectilinearity, are picked up for 3-C stations from **amp3c** and **apma**. These signal features are used in the seismic neural network for determining signal type. The **hydro\_features** table provides the necessary feature measurements for hydroacoustic detections that are used in the hydroacoustic neural network to determine signal type.

## ▼ Operational Procedures

The phase name (*iphase*), station association ID (*stassid*), author (*auth*), and load date (*lddate*) at universal time coordinate (UTC) are updated by *StaPro* in the **arrival** table.

Station and event association information is written to the tables identified by the following four parameters: *assoc-table*, *origerr-table*, *origin-table*, and *stassoc-table*. Their default names are similar to the parameter names.

Other parameters describe the maximum number of records allowed to be returned from the database (*maxrec*) and the database vendor name (*vendor*). The default value for *maxrec* is  $-1$ , which allows all records found to be returned. The default value for *vendor* is `oracle`.

### Signal-type Determination Parameters

Signal-type determination parameters are used in the various methods for determining signal type. The methods include using *CLIPS* rules, the 3-C neural network, or the default rules written in the C language. Typical values used for seismic processing follow. Comment lines begin with a pound (#) sign.

```
## Initial Wave-Type Parameters
## -----
## ab-constant                [def=0.3]
## cfreq1                     [def=-999.0]
## cfreq2                     [def=-999.0]
## cfreq3                     [def=-999.0]
## cfreq4                     [def=-999.0]
## cfreq5                     [def=-999.0]
## def-3c-delaz               [def=20.0]
## def-3c-delslo              [def=5.0]
## max-delslo                 [def=5.0]
## max-noise-velocity         [def=2.9]
## min-teleseismic-velocity   [def=11.0]
## min-p-freq                 [def=3.0]
## min-p-rect                 [def=0.7]
## max-p-hvrat                [def=1.0]
## nnet-log-file              [def=NULL; try $(LOGDIR)/StaPro/$(sta)_nnet.log]
## nnet-weights-file          [def=NULL; try $(STAPRO_SITE_PATH)/ipnnwts.tbl]
## noise-fkqual               [def=5; was called min-fkqual]
```

```

## noise-fkqual-fstat          [def="5.0,3.0"]
## p-s-velocity-threshold      [def=5.7]

max-delslo=999.9
nnet-weights-file=$(STAPRO_SITE_PATH)/ipnnwts.tbl
nnet-log-file=$(LOGDIR)/StaPro/$(sta)_nnet.log

## Phase Grouping Parameters
## -----
## group-amplitude-tolerance    [def=40.0]
## group-delaz-factor           [def=3.0]
## group-first-s-p-max-time     [def=360.0]
## group-s-p-time-factor        [def=2.1]
## local-s-phase                [def=Lg]
## intermediate-p-phase         [def=Pn]
## intermediate-p-vel-boundary  [def=7.5]
## local-delaz-factor           [def=2.5]
## local-max-s-p-time           [def=25.0]
## local-min-p-p-separation     [def=25.0]
## max-grouping-amplitude-factor [def=25.0]
## max-grouping-time-no-azimuth [def=120.0]
## max-p-coda-arrival-time      [def=20.0]
## max-rg-time                  [def=65.0]
## max-teleseismic-amplitude-factor [def=25.0]
## min-grouping-amplitude-factor [def=0.4]
## min-lg-frequency             [def=1.54; also used in Final Phase ID]
## min-teleseismic-amplitude-factor [def=0.4]
## noise-phase                  [def=N]
## teleseismic-amplitude-tolerance [def=40.0]
## teleseismic-delaz-factor      [def=2.0]
## teleseismic-max-p-p-time     [def=90.0]
## unknown-primary-phase        [def=Px]
## unknown-secondary-phase      [def=Sx]
## unknown-teleseismic-phase    [def=tx]
## teleseismic-phase            [def=P]

## Final Phase ID Parameters
## -----
## bayes-file      [def=bayes.tbl; try $(STAPRO_SITE_PATH)/bayes.tbl]
## lg-tol          [def=5.0]
## rg-tol          [def=5.0]
## lg-min-velocity [def=3.1]
## lg-max-velocity [def=5.0]

```

## ▼ Operational Procedures

```

## min-sn-p-time                [def=60.0]
## pg-tol                      [def=3.0]
## phase-distance-file          [def=phase-distance-ranges.txt]
## regional-min-s-p-time        [def=30.0]
## rg-min-velocity              [def=2.8]
## rg-max-velocity              [def=3.7]
## s-split-parm                 [def=0.4]
## sn-confidence-threshold      [def=0.5]
## tt-phases                    [def="'P','Pn','Pg','S','Sn','Lg','Rg'"]
## vmodel-spec-file             [required; try $(VMSF)]
## tl-spec-file                 [required; try $(VMSF)]
## mag-def-file                 [required; try $(VMSF)]
## magtype                      [def=ml]

vmodel-spec-file=$(VMSF)
tl-spec-file=$(TLSF)
mag-def-file=$(MDF)
bayes-file=$(STAPRO_SITE_PATH)/bayes.tbl
phase-distance-file=$(GA_SITE_PATH)/GA_dist_depth_ranges

## Event Confirmation Parameters
## -----
## 3c-azimuth-weight            [def=.25]
## 3c-slowness-weight           [def=.25]
## array-azimuth-weight         [def=0.5]
## array-slowness-weight        [def=0.5]
## event-confirmation-threshold [def=2.6]
## primary-time-weight          [def=1.0]
## secondary-time-weight        [def=0.7]
## semi-major-axis-threshold    [def=400000.0]

```

When *CLIPS* rules are available, Boolean parameters *ipr* (initial-phase rules) and *fpr* (final-phase rules) tell which program states the given rules apply to. The *ipr* flag refers to the signal-type determination processing unit, and the *fpr* flag refers to the phase-identification processing unit.

The neural network parameters *nnet-weights-file* and *nnet-log-file* specify filenames. Both are null by default, which means that these files will not be used by *StaPro*. These two parameters only apply to the seismic 3-C neural network. The parameters related to the 3-C neural network are: *cfreq1*, *cfreq2*, *cfreq3*, *cfreq4*, and *cfreq5*. These five center frequencies are used to constrain the query selecting

the five horizontal-to-vertical amplitude ratio (*htov*) attributes needed from the **amp3c** table. The *htov* are passed into the 3-C neural network to help determine signal type.

Several threshold-type parameters are used in the default rules written in the C language. The parameters are specific to array and 3-C station data. Array parameters place limits on slowness variance, regional phase velocity boundaries, and f-k quality. These parameters include: *max-delslo*, *max-noise-velocity*, *min-p-freq*, *min-teleseismic-velocity*, *noise-fkqual*, *noise-fkqual-fstat*, and *p-s-velocity-threshold*. 3-C parameters limit horizontal-to-vertical amplitude ratios (*max-p-hvrat*) and rectilinearity (*min-p-rect*).

### Signal-grouping Parameters

Signal-grouping parameters are used to group signals resulting from the same hypothetical event. The parameters are organized into categories based on suspected event-station distance as calculated from S-P travel time. The categories are teleseismic, local, or regional groups. Also included are shared parameters used in testing compatibility between two arrivals, miscellaneous parameters that separate intermediate P, unknown phase names, miscellaneous parameters that help identify intermediate P phases (Pg, Pn), and labels for unknown phases.

Teleseismic related parameters include *max-teleseismic-amplitude-factor*, *min-teleseismic-amplitude-factor*, *teleseismic-amplitude-tolerance*, *teleseismic-delaz-factor*, *teleseismic-max-p-p-time*, and *teleseismic-phase*. These parameters define the duration of the time window to use when searching for compatible teleseismic arrivals. Compatibility between two arrivals is based on amplitude and azimuth.

The parameters for local event distance ranges include *local-delaz-factor*, *local-max-s-p-time*, *local-min-p-p-separation*, *local-s-phase*, *min-lg-frequency*, and *max-rg-time*. These parameters define limits on the time separation between local event arrivals (P-to-S and P-to-P coda) when searching for compatible arrivals. Compatibility is based on amplitude and azimuth. The *min-lg-frequency* and *max-rg-time* parameters also help identify Lg from Rg using frequency and arrival-time differences.

## ▼ Operational Procedures

Parameters for regional event distance ranges include *group-s-p-time-factor*, *group-first-s-p-time*, and *max-p-coda-arrival-time*. These parameters define regional limits on the time separation between compatible arrivals (P-to-S and P-to-P coda). Compatibility is based on amplitude and azimuth.

The compatibility test parameters include *group-amplitude-tolerance*, *group-delaz-factor*, *max-group-amplitude-factor*, *max-grouping-time-no-azimuth*, *min-grouping-amplitude-factor*, and *noise-phase*. These parameters define the limits on amplitude and azimuth differences between two arrivals in order to determine if they are compatible.

The intermediate P parameters include *intermediate-p-vel-boundary* and *intermediate-p-phase*. These parameters help distinguish between Pg and Pn phases.

The unknown phase name parameters include *unknown-teleseismic-phase*, *unknown-primary-phase*, and *unknown-secondary-phase*. These parameters provide labels for coda phases.

**Phase-identification Parameters**

The phase-identification parameters are used to identify regional phase names for seismic groups. The parameters are organized into three groups according to the processing that is needed. These include: (1) the Bayesian method, (2) phase prediction, and (3) travel-time model.

The Bayesian method parameters are *bayes-file*, *min-sn-p-time*, *s-split-parm*, *sn-confidence-threshold*, and *regional-min-s-p-time*. These parameters define the Bayesian probabilities that are used to identify a regional S phase based on the time differences (S-P), frequency, velocity, and horizontal-to-vertical power ratio. See [IDC5.2.1] for details on the Bayesian method.

The phase prediction parameters are *lg-max-velocity*, *lg-min-velocity*, *lg-tol*, *pg-tol*, *phase-distance-file*, *rg-max-velocity*, *rg-min-velocity*, and *rg-tol*. These parameters provide limits on Pg, Lg, and Rg velocity ranges and phase-distance relationships.

The travel-time model parameters are *tt-phases* and *vmodel-spec-file* and are used during both phase prediction and event location.

### Event Location and Confirmation Parameters

Event location and confirmation parameters specify event confirmation thresholds used in conjunction with the velocity model specified by *vmodel-spec-file*. The parameters also define the method for estimating local magnitudes and the corrections to use.

The event-confirmation parameters are *array-azimuth-weight*, *array-slowness-weight*, *event-confirmation-threshold*, *primary-time-weight*, *secondary-time-weight*, *semi-major-axis-threshold*, *3c-azimuth-weight*, and *3c-slowness-weight*. These parameters define the weighting values for azimuth, slowness, and time-defining information that are used when determining if an event is above the event-definition threshold.

Two parameters control debug information from *libloc* calls. A verbosity level controlling the volume of messages written to a special log file can be set by *locator-verbose*. The default value is 0, which means no messages are generated. A log filename can be specified in *locator-outfile-name*. A default name of *StaPro\_loc.err* is defined.

The magnitude-related parameters are *mag-def-file*, *magtype*, and *tl-spec-file*. These parameters define the magnitude type found in the magnitude definition file and define the transmission-loss file name.

### Hydroacoustic Parameters

Hydroacoustic detections are classified into three types of phase names. H phases are presumed to be generated by in-water explosions. T phases are generated underground and transmitted to the water column. N phases are noise phases. The parameters for hydroacoustic processing specify these phase name labels and help to determine signal type and groupings. The typical settings for hydroacoustic parameters follow:

## ▼ Operational Procedures

```

## Hydro Parameters
## -----
## h-phase [def=H]
## hf-band-low-cut [def=32.0]
## hf-band-high-cut [def=64.0]
## hydro-look-back-time [def=300.0]
## hydro-nnet-log-file [def=NULL]
## hydro-nnet-weights-file
## [def=NULL; try $(STAPRO_SITE_PATH)/hydro_$(sta).weights]
## lf-band-low-cut [def=3.0]
## lf-band-high-cut [def=6.0]
## n-duration-lim [def=6.0]
## n-num-cross-lim [def=12.0]
## n-phase [def=N]
## n-ratio-lim [def=-15.5]
## n-time-spread-lim [def=35.0]
## n-total-time-lim [def=0.4]
## o-phase [def=O]
## t-num-cross-lim [def=12.0]
## t-phase [def=T]
## t-ratio-lim [def=-15.5]
## t-time-spread-lim [def=5.0]
## unknown-h-phase [def=Hx]
## unknown-o-phase [def=Ox]
## unknown-t-phase [def=Tx]

o-phase="H"
unknown-o-phase="Hx"
hydro-nnet-log-file=$(LOGDIR)/StaPro/$(sta)_nnet.log
hydro-nnet-weights-file=$(STAPRO_SITE_PATH)/hydro_$(sta).weights
hydro-look-back-time=650.0

```

A look-back time parameter called *hydro-look-back-time* is used as a lead and lag time duration that the processing window is extended. This duration should be set in accordance with *DFX*'s maximum processing window. Feature records are identified by high- and low-frequency filter-band parameters *hf-band-high-cut*, *hf-band-low-cut*, *lf-band-high-cut*, and *lf-band-low-cut*. The hydroacoustic neural network runs if a weight's filename is specified in *hydro-nnet-weights-file*. A neural-network log filename can be specified by *hydro-nnet-log-file* (a log file is generated only if



this parameter is set). Phase names are specified by parameters *h-phase*, *n-phase*, *o-phase*, and *t-phase*; however, the hydroacoustic neural network weights are constructed so that they do not classify O phases.

Default rules, written in the C language, are used if the signal type is not determined by the neural network. This is the case, for instance, when a new station comes on-line and no neural-net weights are available for it. Parameters that control thresholds for the default hydroacoustic rules are *n-duration-lim*, *n-num-cross-lim*, *n-ratio-lim*, *n-time-spread-lim*, *n-total-time-lim*, *t-num-cross-lim*, *t-ratio-lim*, and *t-time-spread-lim*.

Hydroacoustic grouping is performed by searching for overlapping signals. Arrivals excluded from a group are given a phase name label according to the parameters *unknown-h-phase*, *unknown-o-phase*, and *unknown-t-phase*.

### Infrasonic Parameters

Infrasonic parameters control the infrasonic processing in *StaPro*, which consists of determining signal type (I or N) and grouping. Typical infrasound parameter settings follow:

```
## INFRA
## -----
## infra-lower-velocity           [def=290.0]
## infra-upper-velocity          [def=660.0]
## infra-look-back-time          [def=300.0]
## infra-unknown-i-phase         [def=Ix]
## infra-i-phase                 [def=I]
## infra-n-phase                 [def=N]
## infra-max-grouping-time       [def=100.0]
## infra-delaz-factor            [def=2.0]

infra-lower-velocity=270.0
infra-max-grouping-time=$(StaPro-infra-max-grouping-time)
```

A look-back time parameter called *infra-look-back-time* is used as a lead and lag time duration that the processing window is extended. It should be set in accordance with the *DFXs* maximum processing window.

## ▼ Operational Procedures

Signal type is determined by comparing the detections velocity to the thresholds in *infra-lower-velocity* and *infra-upper-velocity*. Signals are then grouped together if they are compatible by using the parameters *infra-max-grouping-time* and *infra-delaz-factor*. Phase names are given according to the parameters *infra-i-phase*, *infra-unknown-i-phase*, and *infra-n-phase*.

## ADVANCED PROCEDURES

This section provides detailed instructions for using the software's advanced features.

### Using Neural Networks

A neural network is used in *StaPro* for determining the signal type of seismic (T, P, S, or N) and hydroacoustic (H, T, or N) detections. The inputs that are used in the seismic case include rectilinearity, horizontal-to-vertical ratio, period, and context items describing the frequency of occurrences of detections before and after the detection. The inputs used by the neural network determine the relative likelihood that a detection belongs to each of the possible signal categories. Neural networks must be trained to identify signal types. Feature measures and known signal type are used to train the network. Then, the neural network can process new detections and compute a number that represents the measure of membership the detection has in each signal category [IDC5.2.1]. The hydroacoustic neural network is not the same as the seismic neural network and the weights files differ.

The neural-network weights filenames are set in the *StaPro.par* parameter file (*nnet-weights-file* for seismic and *hydro-nnet-weights-file* for hydroacoustic). If a weight is not set, then the neural network is not used. Similarly, the log-file parameters (*nnet-log-file* for seismic and *hydro-nnet-log-file* for hydroacoustic) must be specified or the log file will not be saved.

Both the seismic and hydroacoustic neural-network weights files must follow prescribed formats, which are described in this section. Format descriptors are used to declare the C language data type being read. The formats include integer (%d), character string (%s), and floating-point number (%f).

Refer to “Tuning Stations” on page 42 for discussion on generating new weights.

### Seismic

The seismic neural network is recommended for use with 3-C stations but not for array stations. There are 16 inputs to the seismic neural network. The inputs consist of feature measurements and two context-based inputs as shown in Table 3.

**TABLE 3: INPUTS TO SEISMIC NEURAL NETWORK**

Number	Attribute	Description
1	<i>arrival.per</i>	inverse of the waveforms frequency
2	<i>arrival.rect</i>	rectilinearity
3	<i>apma.plans</i>	S-phase planarity
4	<i>apma.inang1</i>	long-axis incidence angle
5	<i>apma.inang3</i>	short-axis incidence angle
6	<i>apma.hmxmn</i>	maximum-to-minimum horizontal ratio
7	<i>apma.hvratp</i>	P-phase horizontal-to-vertical ratio
8	<i>apma.hvrat</i>	S-phase horizontal-to-vertical ratio
9	<i>ddet</i>	context: difference between number of detections before and after the detection
10	<i>dtime</i>	context: difference between average time residual of detections before and after the detection
11	<i>amp3c.htov</i>	horizontal-to-vertical amplitude ratio at 0.25 Hz
12	<i>amp3c.htov</i>	horizontal-to-vertical amplitude ratio at 0.5 Hz
13	<i>amp3c.htov</i>	horizontal-to-vertical amplitude ratio at 1.0 Hz
14	<i>amp3c.htov</i>	horizontal-to-vertical amplitude ratio at 2.0 Hz
15	<i>amp3c.htov</i>	horizontal-to-vertical amplitude ratio at 4.0 Hz
16	<i>arrival.slow</i>	slowness value (only for array station data)

## ▼ Operational Procedures

The context parameters are important because they provide additional information that a seismologist would use when examining a waveform. A seismologist intuitively notices if more disturbances follow a detection than precede it. This helps determine if the detection is P or S. This context information is expressed by calculating the difference between the number of disturbances after and before the detection. This information is calculated using a fixed time window specified in the neural-network weights file and is typically 60.0 seconds. If *ddet* (from Table 3) is greater than zero then the signal type is likely to be P; if *ddet* is less than zero, then the signal type is likely to be S. A similar measure is calculated for the average time residual of detections in the time window relative to the detection being evaluated (*dtime* in Table 3). The *dtime* measure provides more weight to detections close in time over those occurring earlier or later than the detection being evaluated.

The *arrival.slow* attribute rarely is used because it is only included if processing an array station via the neural network. All attributes listed in Table 3 are normalized to keep their values in a range between -1 and 1.

The seismic neural network is run in three stages, where each stage has two possible outcomes. The first stage separates signal and noise. The second stage separates primary and secondary signals. The third stage separates teleseismic P from local and regional P. A set of weights must exist for each of the three stages.

The seismic weights file has traditionally contained neural-network weights for 3-C stations. The weights file used by *StaPro* consists of ancillary information in addition to the weights. Table 4 describes the general format of the seismic neural-network weights file. The weights in the following example show the ancillary information and stage-one weights for station STKA. A comment line is any line with "\*" in the first column. See [IDC5.2.1] for a description of the seismic neural-network design.

```
*-- Example section from seismic weights file.
1
*-- STKA weights for stage 1 neural network begin here.
STKA
3
60.000000
0.25 0.50 1.00 2.00 4.00
```

STKA TPS-N

628 3 16 7 2

-14.763854 -16.715521 -3.25238 -34.219540 -10.764621 -8.567822

0.567249 -1.904365 0.085101 1.648242 2.902455 2.399817

14.228270 14.150584 18.506845 19.109190 5.858047 8.834391

1.121436 -20.997845 4.488671 9.672775 -9.155566 -18.381361

-7.443598 7.778023 -7.084640 5.762690 15.863204 4.954304

4.852483 9.826913 -10.590254 -0.601912 -4.480463 12.378825

0.380726 15.311364 -7.977109 -2.982994 2.752584 -2.894454

-9.454600 -2.019959 5.599975 -10.095946 -3.764777 -1.840247

-6.915637 0.568362 13.358426 -12.458193 -6.887399 7.256957

25.900608 -6.286942 -1.067941 11.901111 31.475147 -25.096247

-8.519755 -3.435860 3.520229 0.734010 -0.859587 3.194686

-5.553283 4.912724 -2.365806 -3.639076 3.277486 0.846160

2.627199 -16.306238 -3.868679 4.505494 1.223432 1.716734

4.886774 8.081739 -6.303213 -11.449767 -2.193998 4.363800

15.331384 -16.401327 -1.851413 -1.018845 -5.042238 5.990855

-9.512029 19.646658 2.434661 -5.217672 -3.200002 8.002850

-40.707901 40.707909

16.311863 -16.311859

15.686264 -15.686270

32.753483 -32.753490

20.709629 -20.709637

21.806301 -21.806303

34.330822 -34.330841

25.428436

0.500000

0.505 0.515 0.525 0.535 0.545 0.555 0.565 0.575 0.585 0.595 0.605

0.615 0.625 0.635 0.645 0.655 0.665 0.675 0.685 0.695

0.5976969 0.4867996 0.3770624 0.3490854 0.3521079 0.4214766

0.4963668 0.5708284 0.6283861 0.6960425 0.7512097 0.8151730

0.8560692 0.8823753 0.9086716 0.9388899 0.9706593 0.9899895

0.9946248 0.9987423

0.4853042 0.5188849 0.5555364 0.6276949 0.7230064 0.7812760

0.8288509 0.8678071 0.9065589 0.9311690 0.9531009 0.9430134

0.9230905 0.9006768 0.8623297 0.8341841 0.8088839 0.7978647

0.7903540 0.7834764

\*--

## ▼ Operational Procedures

*StaPro* loops over the stations whose neural-network weights are contained in the file (item 1 from Table 4) reading the individual station weights information. The second through fifth items in the file contain the station's ancillary information such as station name, context window length, center frequencies, and size of the network.

Weights for the input node side of the hidden layer are read during a loop (item 8). This multi-line input has  $N$  rows and  $M-1$  columns, where  $N$  is the number of input nodes, and  $M$  is the number of hidden layers.

**TABLE 4: FORMAT OF SEISMIC NEURAL NETWORK WEIGHTS FILE**

Items	Format	Description
1	%d	number of stations represented in the file
2	%s	station name
3	%d	station type: 1 for array and either 2 or 3 for 3-C stations
4	%f	length of the context window
5	%f %f %f %f %f	five space-delimited center frequencies
6	%s	title line (not processed by <i>StaPro</i> )
7	%d %d %d %d %d	number of iterations used during training number of network stages (3 expected) number of input nodes (features) number of nodes in hidden layer number of output nodes (signal type categories)
8	%f	weights corresponding to nodes between the input and hidden layers
9	%f	weights corresponding to nodes between the hidden and output layers
10	%s	criterion function value
11	%f	threshold for choosing output node 1 over node 2

**TABLE 4:   FORMAT OF SEISMIC NEURAL NETWORK  
WEIGHTS FILE (CONTINUED)**

Items	Format	Description
12	%f	bins that have confidence values tied to them for each output node (the length is 20 values)
13	%f	confidence values for output node 1 corresponding to bin values in item 12
14	%f	confidence values for output node 2 corresponding to bin values in item 12

Similarly, weights for the output-node side of the hidden layer are also read in a loop (item 9). In this case,  $N$  is the number of nodes in the hidden layer and  $M$  is the number of output nodes. The total number of weights read for the output layer is  $N \times M$ .

Confidence values for the two output nodes are specified in two vectors that span the bin range specified in a third vector (items 12, 13, and 14). The length of these vectors is fixed at 20. Items 10 and 12–14 currently are not used by *StaPro*.

Weights for each neural-network stage (signal versus noise, primary versus secondary signals, and teleseismic P versus regional P) are stored in items 6–14.

### Hydroacoustic

This section discusses the hydroacoustic neural network and weights file. The hydroacoustic neural network is designed to take a variable number of input features. The exact number depends on the features and frequency pass-bands used to train the network. Up to 20 features can be measured in up to 10 pass-bands. The features and bands are measured by *DFX* and stored in the **hydro\_features** database table. All features are read into *StaPro* and stored in a two-dimensional (2-D) array, and the weights are read for the station being processed. Only those feature-bands used to train the station are used in the neural-network

## ▼ Operational Procedures

evaluation. Table 5 lists the available features. Frequency pass-bands are stored in **hydro\_features**.*low\_cut* and *high\_cut* and consist of 2–4, 2–80, 3–6, 4–8, 6–12, 8–16, 16–32, and 32–64 Hz.

**TABLE 5: INPUTS TO HYDROACOUSTIC NEURAL NETWORK**

Number	Attribute	Description
1	<i>peak_time</i> (PT)	<i>peak_time</i> – <i>onset_time</i>
2	<i>peak_level</i>	<i>peak_level</i> – <i>total_energy</i> + 10 *LOG10(TT)
3	<i>total_energy</i>	PT - MAT / TT
4	<i>mean_arrival_time</i> (MAT)	<i>mean_arrival_time</i> – <i>onset_time</i>
5	<i>time_spread</i>	<i>time_spread</i>
6	<i>onset_time</i>	<i>total_time</i> / TT
7	<i>termination_time</i> (TT)	<i>termination_time</i> – <i>onset_time</i>
8	<i>total_time</i>	<i>num_cross</i> / TT
9	<i>num_cross</i>	<i>num_cross</i>
10	<i>ave_noise</i>	proportional to sum of TT, <i>total_energy</i> , and <i>ave_noise</i>
11	<i>skewness</i>	<i>skewness</i>
12	<i>kurtosis</i>	<i>kurtosis</i>
13	<i>cep_var_signal</i>	<i>cep_var_signal</i>
14	<i>cep_delay_time_signal</i>	<i>cep_delay_time_signal</i>
15	<i>cep_peak_std_signal</i>	<i>cep_peak_std_signal</i>
16	<i>cep_var_trend</i>	<i>cep_var_trend</i>
17	<i>cep_delay_time_trend</i>	<i>cep_delay_time_trend</i> .
18	<i>cep_peak_std_trend</i>	<i>cep_peak_std_trend</i>
19	<i>extra</i>	<i>ave_noise</i>
20	<i>sig_in_band</i>	10 if TT ≤ 0 and features are available for that band; else 100



The hydroacoustic neural network is run in two stages. Stage 1 separates T-phase and noise (N) or in-water (H) disturbances; stage 2 separates noise and in-water events. A set of weights must exist for each of the stages.

Each hydroacoustic station has a separate neural-network weights file, unlike the all-in-one weights file for seismic stations. The weights file used by *StaPro* consists of ancillary information in addition to the weights. Table 6 on page 35 describes the format for the hydroacoustic weights file. An example of an abbreviated weights file for station PSUR follows. Stage 2 weights are not shown in the example.

```

STAGE 1 NNET: psurNTO_1.hnet HN/HO-target HT-clutter
TRAINING SET: psurNTO.clu
  14 FEATURES SELECTED TMD =    0.200000 TCC =    0.900000
  5 COMPONENTS AT tpc =    85.0000
BANDS[ 8] 2- 4   2- 80   3- 6   4- 8   6- 12   8- 16   16-
32 32- 64
LOG
    0.42549    -1.29493    2.71228

----- 158 rows of item 4 (see Table 6) removed for this example -----

    0.86859    -3.00000    2.87050
    1.60000    -0.80000

14
36 CEP_VAR_TREND band 8: 2-80 hz                0.218822
25 TIME_SPREAD (sec) band 8: 2-80 hz            0.266140
61 PEAK_DELAY (sec) band 3: 4-8 hz              0.295965
41 PEAK_DELAY (sec) band 2: 3-6 hz              0.369845
105 TIME_SPREAD (sec) band 5: 8-16 hz           0.395263
140 SIGNAL_IN_BAND6 band 8: 2-80 hz             0.408401
 4 MEAN_DELAY (sec) band 1: 2-4 hz              0.419903
84 MEAN_DELAY (sec) band 4: 6-12 hz             0.508785
64 MEAN_DELAY (sec) band 3: 4-8 hz             0.650867
44 MEAN_DELAY (sec) band 2: 3-6 hz             0.684057
87 DURATION (sec) band 4: 6-12 hz              0.805216
160 SIGNAL_IN_BAND7 band 8: 2-80 hz            1.07478
67 DURATION (sec) band 3: 4-8 hz              1.27469
45 TIME_SPREAD (sec) band 2: 3-6 hz            1.47251

TAN
    2.000000    -0.606615    0.321887    0.616578    0.498727
15 5

```

## ▼ Operational Procedures

16	
15	
1	0.2061
2	-0.3738
3	-0.0813
4	1.0913
5	0.4701
6	0.2737
7	0.6275
8	-0.4480
9	0.4327
10	0.6491
11	0.0935
12	0.2014
13	-0.9029
14	0.0594
15	-1.3026

----- 4 blocks of items 11 – 14 (see Table 6) removed for this example -----

1	
21	
6	
1	0.2497
16	0.6461
17	-0.0735
18	-0.9541
19	0.1282
20	-0.7411

**TABLE 6: FORMAT OF HYDROACOUSTIC NEURAL NETWORK WEIGHTS FILE**

Items	Format	Description
1	%s	four title lines are read but not processed
2	%s and %2d and %3d-%3d	pass-band information; the first five characters must contain the string BANDS; positions 7–8 define the number of bands to be read; after position 11, the lower- and upper-corner frequencies are read in a loop for all bands
3	%s	input normalization type is LOG
4	%f %f %f	input normalization factors are read in a loop
5	%f %f	two output normalization factors
6	%d	number of inputs (feature-band combinations) to be read (referred to as Ninputs)
7	%d	indexes of feature bands used in training (loop over Ninputs)
8	%s	transfer type of TAN
9	%f	loop reads confidence band, mean of clutter distribution, clutter standard deviation, mean of target distribution, and target standard deviation
10	%d %d	number of input nodes with one bias node included and number of hidden layers nodes
11	%d	PE index of hidden layer
12	%d	number of weights to read between input and hidden layers
13	%s	blank line
14	%d %f	weights node index and value for hidden layer
15	%d	number of output nodes
16	%d	PE index of output layer

## ▼ Operational Procedures

**TABLE 6: FORMAT OF HYDROACOUSTIC NEURAL NETWORK WEIGHTS FILE (CONTINUED)**

Items	Format	Description
17	%d	number of weights to read between hidden and output layers
18	%s	blank line
19	%d %f	weights node index and value for output layer

Six feature measurements and a combination of pass bands account for the 14 inputs used to generate the PSUR weights. Pass bands (item 2) used during neural-network training are specified in a complicated read format. The order of the bands are important because it dictates the index locations that map feature-bands with the proper weights (see item 7).

Input normalization factors (item 4) are read three-to-a-line. A loop is required to read all of the factors. The loop repeats  $N$  times where  $N$  is the product of the number of bands (8 for PSUR; see item 2) and the maximum number of features allowed (20; defined in the software).

The indexes (item 7) of the feature-bands used to train the network are read one-per-line. The indexes point to the corresponding feature-bands to be used as input to the hydroacoustic neural network evaluator.

The number of hidden layer nodes (item 10) and number of hidden layer weights to be read (item 12) define inner and outer loops that allow blocks of weights (item 14) for the hidden layer to be read. A similar set of loops is needed to read weights (item 19) for the output layer.

Stage 2 weights are read using the format for the 19 items described above.

## Using CLIPS Rules

Station-specific rules are an optional way of determining signal type and/or identifying phases for seismic stations (see Detailed Design in [IDC7.1.12]). This method is invoked by specifying a file name in the *StaPro* parameter *sta-rule-file*. This file contains the necessary rules to determine the signal type/phase for each detection. The rules are written in the *CLIPS* Expert System (interpretive) language. They consist of If-Then block statements that are “fired” simultaneously. This section describes *CLIPS*-related parameters, functions, facts, and selected examples of rules.

Two Boolean parameters (*ipr* and *fpr*) define the program state in which the *CLIPS* rules are used. Setting *ipr* to true indicates that some of the rules apply to determining signal type; setting *fpr* to true indicates that rules apply to identifying phases.

A selection of functions written in the C language and bound to *CLIPS* are available in *StaPro*. For example, the seismic neural network can be called from *CLIPS* rules. Table 7 shows a list of available functions. The list includes items related to determining signal type and identifying phases. Several range-check functions are available to verify that database attribute values are within the allowed ranges as specified in [IDC5.1.1Rev2].

**TABLE 7: STAPRO FUNCTIONS BOUND TO CLIPS**

Function Name	Program State(s)	Input Arg(s)	Output	Description
init-phase-nnet	ipr	none	none	runs seismic neural network and inserts facts (nnet-status <i>stat</i> ) where 0 = success and 1 = failure, and (nnet-phase <i>iwt</i> ) where <i>iwt</i> is T, P, S, or N
save	ipr	<i>sig_type</i> , ( <i>not_p</i> , <i>not_t</i> , <i>not_s</i> , <i>not_n</i> )	none	saves signal type (T, P, S, or N) in internal data structure member (.iwt) if allowed by optional Boolean NOT flags

## ▼ Operational Procedures

TABLE 7: STAPRO FUNCTIONS BOUND TO CLIPS (CONTINUED)

Function Name	Program State(s)	Input Arg(s)	Output	Description
verbose-check	ipr, fpr	<i>level</i>	True or False	controls verbosity level (integer) for output messages compared to <i>verbosity</i> parameter; the higher the level the more extensive the messages
iwt-by-velocity	ipr	none	True or False	uses loop counter to indicate current detection; returns true if <i>slow</i> and <i>delslo</i> are in range and <i>delslo</i> < <i>max_delslo</i> ; otherwise it returns false
noise-screener	ipr	none	True or False	returns true (is noise) if <i>fkqual</i> and <i>fstat</i> are in range and either <i>fkqual</i> ≥ <i>noise_fkqual</i> or ( <i>fkqual</i> ≥ <i>noise_fkqual_fstat[0]</i> and <i>fstat</i> < <i>noise_fkqual_fstat[1]</i> ) is true; otherwise it returns false
check- <i>attrib</i> -range	ipr	<i>attrib</i> <sup>1</sup>	True or False	checks range of attribute
get-largest-s	fpr	none	True or False	inserts feature measurement facts about S with largest amplitude from working group and inserts control fact (largest-S-facts)
final-save	fpr	<i>arid</i> , <i>iphase</i>	none	saves phase name in internal data structure member ( <i>.iphase</i> )

1. Range checking functions are available for *rect*, *hvrat*, *fstat*, *fkqual*, *per*, *slow*, *delslo*, *azimuth*, and *delaz*.

Facts are loaded during initialization before they can be evaluated in rules. In *StaPro*, facts consist of feature measurements, user parameters, and control information and they are examined by the rules and are used in conditional statements that control whether or not a rule is fired. A rule may adjust the fact list by adding or deleting facts as needed. Facts have two parts: an attribute and a value. Both are enclosed in parentheses such as (*attribute value*).

Control information provides facts describing the stage of processing and whether or not important data files are available, such as the neural-network weights. This type of information is determined within *StaPro*. Different control information is needed when determining signal type (ipr) versus when identifying phases (fpr). Table 8 lists the control facts that are available for the ipr and fpr program states and describes their usage.

Selected user parameters and feature measurements are also available on the fact list. Most of these are available during both program states (ipr and fpr). An example parameter fact that specifies the maximum allowed noise velocity is (*max-noise-velocity value*) where *value* is replaced by the *max-noise-velocity* parameter within *StaPro*.

**TABLE 8: CONTROL FACTS AVAILABLE TO CLIPS RULES**

State	Fact	Description
ipr	(program-state initial-wave-type)	determining signal type state
ipr	(ans-wts)	neural network weights available where <i>ans</i> is yes or no
ipr	(not-iwt 0)	Boolean not-allowed flag for signal type of <i>iwt</i> where <i>iwt</i> is t, p, s, or n; true (1) means this signal type is not allowed; false (0) means it is allowed
fpr	(program-state final-phase-id)	determining phase name state
fpr	(ngp num)	<i>num</i> is the number of members in a group
fpr	(piwt P)	signal type is P

## ▼ Operational Procedures

TABLE 8: CONTROL FACTS AVAILABLE TO CLIPS RULES (CONTINUED)

State	Fact	Description
fpr	(siwt S)	signal type is S
fpr	(iwtiphas -)	iwt phase name (P or S) not determined
fpr	(iwtvel V)	V is the velocity of the iwt signal type (P or S)
fpr	(sminusp DT)	DT is the time difference between S and P
fpr	(largest-S-facts)	inserted as result of get-largest-s function

An example of a database fact for the onset time of a detection is (*time value*), where *value* is replaced by *arrival.time* from the database. However, when identifying phases (fpr) for regional seismic groups, both the P-wave and S-wave features are examined simultaneously. For this reason, the database attributes have a prefix of either *p* or *s*, such as (*ptime value*) and (*stime value*).

See the *StaPro* man page for the latest information about *CLIPS* facts.

Alternative methods to determine signal type in *StaPro* can be used if the station-specific rules fail. The neural network may be applied and/or the default rules written in the C language may be applied.

Identifying phases can only be partially completed through user-controlled *CLIPS* rules. The goal of the rules is to determine the first P and/or first/largest S in the group. A phase name for any member of a group may be left undetermined. The default rules written in the C language run after the users' rules and complete the determination. However, the default rules do not override any phases determined by the users' rules. They supply phase names for unidentified arrivals either by Bayesian analysis or prediction.



The general format of a *CLIPS* rule follows:

```
(defrule myrule
  ...conditions...

=>

  ...action...)
```

All *CLIPS* rules specify what “action” is to be taken provided all “conditions” are met. See the *CLIPS* manual for details on writing rules in the *CLIPS* language [Ril98]. An example of a *CLIPS* rule that assigns the Rg label to an S arrival follows:

```
;; This example labels the S member of a group as Rg
;; if the hypothetical event occurred within a defined
;; area based on azimuth and S minus P time.
;;
(defrule Label-S-Rg-Rule                                ;;
  (program-state final-phase-id)                        ;;
  (pazimuth ?paz)                                       ;;
  (pdelaz ?pdaz)                                        ;;
  (sminusp ?sp)                                         ;;
  (sarid ?sarid)                                        ;;
  (test (in-area Largest-S-Rule ?paz ?pdaz ?sp))      ;;
  (test (check-azimuth-range ?paz))                   ;;
  (test (check-delaz-range ?pdaz))                   ;;
=>                                                       ;;
  (final-save ?sarid Rg)                                ;;
  (if (verbose-check 1) then (printout t "Label-S-Rg-Rule" crlf)))
```

A program state helps define when a rule can be fired. In this example, the rule fires during phase identification of arrivals that have been grouped. Additional tests are done in this example to verify azimuth and delaz ranges and to determine if the S arrival is from a particular area. This last test is based on S minus P travel time, azimuth, and its variance. If all conditions are met, then the S arrival is labeled Rg and saved.

## ▼ Operational Procedures

**Tuning Stations**

Station tuning is required when a new station is added to the network so that *StaPro* can process the new data. Tuning consists of setting special parameters, adding database records, and creating new data files for the station. Many of the items discussed here are required. Optional and suggested changes are also included. The changes are grouped into two categories representing general and *StaPro*-specific changes. Tables 9 and 10 describe these categories of changes, respectively. The data type column in these tables describes seismic (S), hydroacoustic (H), and infrasonic (I) stations. Other changes are required for “up-stream” processes such as *DFX*, and it is assumed that *DFX* has populated **arrival** for the new station.

**TABLE 9: GENERAL CHECKLIST FOR ADDING NEW STATION**

Data Type	Conditions	Item	Description
S/H/I	required	site	add new station record to database table
S/H/I	required	shared.par	add new station name to processing list
H/I	required	STA.par	create new station par file to define <i>StaPro</i> <i>statype</i>
S	optional	STA.par	can define station-specific parameters here
S/I	required	sasc.STA	create new station SASC file to define additional modeling errors for <i>delaz</i> and <i>delslo</i>
S	optional	VMSF	can define region-specific velocity model if available
S	optional	TLSF	can define region-specific transmission-loss model if available

TABLE 10: STAPRO-SPECIFIC CHECKLIST FOR ADDING NEW STATION

Data Type	Conditions	Item	Description
S	suggested	<code>ipnnwts.tbl</code>	should provide seismic neural network weights for new 3-C station
H	suggested	<code>hydro_sta.wts</code>	should provide hydroacoustic neural network weights for new station
S	suggested	<code>bayes.tbl</code>	should add station-specific Bayesian probabilities once available

Several general changes are required when a new station is added (see [IDC6.2.4] for more information). For all data types the new station must have a **site** record added, and the station name must be added to the processing list in `shared.par`. Hydroacoustic and infrasonic stations must have the *StaPro* parameter *statype* defined in a station par file. Seismic stations do not require a *statype* value because they use the station type defined in **site**. The station par file may also define station-specific parameters (see “Basic Procedures” on page 15). Seismic and infrasonic stations must have SASC files that define an additional modeling error for *delaz* and *delslo*. A region-specific velocity model and transmission-loss model can be specified for seismic stations if available. Otherwise, the standard IASPEI velocity model and default magnitude coefficients are used.

Neural-network weights should be provided for any new 3-C seismic or hydroacoustic station (see “Neural-network Weights” on page 44). Generate the weights and test them off-line before introducing them into operations. If weights are not provided, then default rules in the code are used to determine signal type in *StaPro*.

Station-specific Bayesian probabilities can also be added for seismic stations. They are used to determine the *S<sub>n</sub>* arrival in the group. The probabilities are determined using data collected from the new station. Because the amount of data from a new station may be limited, a default set of probabilities is available. The default set is based on the probabilities for the station NORESS.

## ▼ Operational Procedures

**Neural-network Weights**

Neural-network weights should be provided for all 3-C and hydroacoustic stations. In some cases, weights from another station may be used as long as the performance is satisfactory compared to that obtainable by the default rules. For example, weights for the 3-C station STKA have been used successfully at station ZAL for some time. It may also be acceptable to use another station's weights for a new station until enough ground-truth information has been acquired to train the neural net to obtain station-specific weights. Ground-truth consists of a well-distributed population of previously identified signals that includes teleseismic (T), regional/local primary (P), secondary (S), and noise (N) categories. At the PIDC, weights for 3-C stations have been trained using ground-truth derived from the Reviewed Event Bulletin (REB). An adaptive training approach makes use of analyst-verified signals to derive station-specific weights (see [Wan97a] [Wan98c]). Performance using the new weights shows favorable results.

Hydroacoustic neural-network weights have been trained for stations PSUR and WK30. The weights for WK30 are also used at station WK31. Additional weights should be generated for other hydroacoustic stations. Default rules are being used at stations that do not have weights specified.

Neural-network COTS software may be used to derive the network weights. One example is NeuralWare Professional II by MathWorks, which was used to calculate the hydroacoustic weights.

**MAINTENANCE**

*StaPro* is a low-maintenance program that writes log files. In the case of 3-C stations, it generates multiple log files for each 3-C station processed, and these files need to be maintained. Otherwise, a single log file is generated for each station processed. No temporary tables are written by *StaPro*.

## Maintaining Log Files

*StaPro* writes messages to the standard error device. These messages describe intermediate results from processing steps and/or error states that may have occurred. The messages form the content of the primary *StaPro* log file. Station-specific log files are organized in a directory structure like `$(LOGDIR)/YYYYDDD/StaPro/$(STA)`. Create the necessary directory structure and remove old log files after a pre-determined number of days.

Processing of 3-C and hydroacoustic stations may also generate neural-network log files. The file path and name is specified by the *StaPro* parameters discussed in “Basic Procedures” on page 15. Station-specific neural-network log files are overwritten for each interval processed. The neural-network log files are typically stored in `$(LOGDIR)/StaPro/$(STA)_nnet.log`. No routine maintenance is required of these log files. But, you must create the directory structure for the neural-network log files.

## SECURITY

### Passwords

A database password is needed to make connection to the station processing database. This string is stored in the system-level parameter file called `$(CMS_CONFIG)/system_specs/process.par`. This file is protected, so only the operator has read and write privileges.



## Chapter 3: Troubleshooting

This chapter describes how to identify and correct problems related to *StaPro* and includes the following topics:

- Monitoring
- Interpreting Error Messages
- Solving Common Problems





## Log Monitoring

The messages returned during *StaPro* processing contain many details about every interval processed. If an interval fails, review the log file to obtain an understanding of the problem. Log files are useful even when an interval is successful because they describe aspects of the decision process used for the interval.

## INTERPRETING ERROR MESSAGES

*StaPro* log files are saved in IDC Operations in station-specific file names. If *StaPro* is being run in a catch-up mode, then the logging information from multiple *StaPro* instances could be mixed within one file, thus making it difficult to sort the messages from each *StaPro* run.

When an error occurs, *StaPro* processing prints a message describing the situation and function name. It then returns an exit value of 1 to the *tuxshell* program. The following section describes some of the most common errors that *StaPro* can recognize.

### Error Exit Messages

---

Message:   Error in db\_initialize:   gdi\_open failed

Description: A problem occurred while initializing the database.

Action:     Use SQL\*Plus to verify that the database can be accessed, and verify that the GDI shared object file is accessible.

---

Message:   Error in clips\_initialize: Error station specific  
            rule file *filename*

Description: A problem initializing the rule file (*filename*) occurred.

Action:     Examine the rule file *filename* for an improperly formatted rule.

---

## ▼ Troubleshooting

---

Message: Error in main: STAPRO EXITING after call to  
select\_detection\_data()

Description: An error occurred when querying the database for detection characteristics.

Action: Check that the necessary database tables are accessible.

---

Message: Error in get\_site\_index: Trouble finding station  
index in site array!

Description: A problem was encountered while obtaining the *site* record for the station of interest.

Action: Verify that station is in the **site** table and that the table is accessible to *StaPro*.

---

Message: Error in main: STAPRO EXITING after check for arids  
with no iphase

Description: The *iphase* attribute was incorrectly left as '-' after *StaPro* processing.

Action: Examine the station's log file for clues why no *iphase* was determined.

---

Message: Error in main: STAPRO EXITING after call to  
clear\_stassids()

Description: The maximum number of *stassids* was reached.

Action: Reduce the processing interval size so that fewer than 254 *stassids* are processed.

---

---

Message: Error in main: STAPRO EXITING after call to iwt()

Description: An error occurred while the default rules were determining the initial wave type.

Action: Examine the *StaPro* parameters and arrival characteristics for inconsistent values.

---

Message: Error opening <hydro\_nnet\_logfile>

Description: Weights for this station are not available.

Action: Either provide station weights in designated file or rely on the default rules coded in *StaPro*.

---

Message: Error in main: STAPRO EXITING after calling locate()

Description: An internal error occurred while forming a *stassid* group, calculating a single-station magnitude, or mapping confirmed events into a data structure.

Action: Examine location and magnitude parameters and arrival characteristics for inconsistent values.

---

Message: Error in main: STAPRO EXITING after calling unchanged\_arrivals()

Description: An error occurred while looking for unchanged arrivals.

Action: None. An internal list comparison was violated.

---

## ▼ Troubleshooting

---

Message:    Error in main: STAPRO EXITING after calling  
                 update\_database()

Description: A problem occurred while trying to update **arrival** records.

Action:      Use SQL\*Plus to access the **arrival** table and verify that arrival identifiers (*arid*) exist for the *StaPro* processing segment.

---

Informative messages are printed by *StaPro* to give the user a sense of the *StaPro* decision process, which is driven by user parameters, availability of external data (for example, neural network weights), and data type. The following section describes examples of the most common informative messages.

**Non-exit Messages**


---

Message:    no neural network weights found for sta *station*

Description: Neural-network weights for this station (*station*) are not available.

Action:      Either provide station weights in the designated file, or rely on the default rules coded in *StaPro*.

---

Message:    Error opening *nnet\_logfile* Revert to Default Rules

Description: The neural-network log file (*nnet\_logfile*) or path does not exist or is not accessible to *StaPro*.

Action:      Manually create *StaPro*'s neural-network log directory, or rely on the default rules rather than the neural network to determine initial wave type.

---

---

Message:   Warning 0 in select\_detection\_data: no data found in database to process

Description: No arrivals were found in the specified window to process.

Action:     None.

---

Message:   No arrivals need updating.

Description: The arrivals for this *StaPro* processing segment were already processed during an adjacent *StaPro* processing segment. The arrival *iphase* and *stassid* information are current.

Action:     None.

---

Message:   Un-Changed Arrivals Are Removed=0 Now, ndet=7

Description: In this example, all seven arrivals have *StaPro* processing results, which are not currently reflected in the **arrival** table. These results will be updated in the database.

Action:     None.

---

A separate neural-network log file will be created if the neural network is used in *StaPro* processing. Typically, a separate log file is used for each station. The log contains messages for each arrival processed. Normally, the activation values determined by the neural network are written to this file, but error messages are written instead if any polarization attributes are out of range. An example of this error message follows:

## ▼ Troubleshooting

---

Message: 31598863: invalid htov1 = -999.0

Description: The horizontal-to-vertical ratio was out of range for this arrival. *StaPro*'s internal default rules were used instead of the neural network.

Action: Investigate why this polarization attribute is invalid. Rerun *StaPro* on this station interval if the problem can be corrected.

---

## SOLVING COMMON PROBLEMS

This section provides instructions for solving common problems that occur in the software.

If the GDI shared object is not found, then *StaPro* exits with an error. The path to this file is specified by the environment variable *GDIHOME*.

Typically, the neural network is used to determine the initial signal type of 3-C stations. *StaPro* uses the neural network as long as a neural network log directory exists, weights are available, and polarization attributes have valid ranges. Error messages are written into the primary *StaPro* log file if the log directory specified does not exist, or no neural-network weights can be found for the station. Error messages are written into the neural-network log file for arrivals with invalid polarization values. In either of these cases, the default rules are implemented instead of the desired neural network.

If no neural-network weights are available for a station, then you can use weights from another station. Choose a station that has geological characteristics similar to the new station. All 3-C stations should use the neural network procedure when possible. See the description of the neural-network weights format in Table 4 on page 30.

## Error Recovery

*StaPro* does not require special clean-up of database records to rerun a time segment. You may adjust a configuration file and simply rerun *StaPro* on a time segment that has already been processed. For example, if *StaPro* failed because the neural network log directory did not exist, then you should create the log directory and rerun *StaPro* on the failed segment. In this case, the neural network would now be used and could produce different results. These new results would automatically be updated in the database to reflect the use of the neural-network processing.





## Chapter 4: Installation Procedures

This chapter provides instructions for installing the software and includes the following topics:

- Preparation
- Executable Files
- Configuration Data Files
- Database
- Tuxedo Files
- Initiating Operations
- Validating Installation

## Chapter 4: Installation Procedures

### PREPARATION

For automatic processing, *StaPro* is typically installed as part of the broader installation of the automatic processing pipeline. This description assumes that the Distributed Applications Control System [IDC6.5.2Rev0.1] has already been installed and configured.

#### Obtaining Released Software

Obtain the software via FTP from a remote site or via a physical medium, such as tape or CD-ROM. Store the software and associated configuration data files as one or more tar files in an appropriate location on a local hard disk. Untar the tar files into a standard UNIX directory structure.

#### Hardware Mapping

You must select the hardware on which to run the software components. Software components are generally mapped to hardware to be roughly consistent with the software configuration model. *StaPro* is typically executed on the same machine used for signal processing with *DFX*.

#### UNIX System

*StaPro* requires the configuration data files to be stored on the UNIX filesystem in the standard configuration tree.

## EXECUTABLE FILES

The single *StaPro* binary must be copied into the `$(RELBIN)` directory.

## CONFIGURATION DATA FILES

Configuration data files for *StaPro* are stored in the UNIX filesystem in a specific directory structure. The root directory of this tree is defined in your environment as `CMS_CONFIG`. This variable is used to reference subdirectories of the configuration. The majority of *StaPro*-related parameters are defined in file `$(CMS_CONFIG)/app_config/automatic/StaPro/StaPro.par`. This file relies on another file called `$(CMS_CONFIG)/system_specs/process.par` for site-specific parameter settings such as the database and log directory names. Additional information for hydroacoustic and infrasonic data is provided by station-specific parameters in the `$(CMS_CONFIG)/station_specs/$(STA).par` files. Additional data files unique to *StaPro* are installed in the `$(CMS_CONFIG)/earth_specs/STAPRO/` directory. These files hold neural-network weights for seismic and hydroacoustic stations and Bayesian inference information used for determining final phase names of regional S arrivals.

*StaPro* relies on several data files that are installed in other `$(CMS_CONFIG)/earth_specs` directories. The GA distance-depth-ranges file is created during installation of the GA Subsystem [IDC6.5.12]. Travel-time, transmission-loss specification, magnitude definition, and slowness-azimuth station correction data files are also referenced by *StaPro* and must be installed accordingly.

## DATABASE

This section describes database elements required for operation of this software component including: accounts, tables, and initialization of the `lastid` table.

### Accounts

*StaPro* uses the IDCX database account. This is the only account required by *StaPro* [IDC5.1.3Rev0.1].

## ▼ Installation Procedures

Special studies that use *StaPro* processing can use an off-line database account, provided the necessary tables are available.

**Tables**

*StaPro* does not require special tables. See Table 1 on page 7 for a list of the *StaPro* database inventory.

**Initialization of lastid**

The **lastid** database table contains the names of identifiers (*keyname*) and identifier values (*keyvalue*) for many tables in the database schema [IDC5.1.1Rev2]. The *keyvalue* contains the value of the *keyname* to use when inserting a new row in the table. The *keyvalue* ensures that the new row is unique. Starting values for these keys should be greater than zero and as small as possible, but must not conflict with existing values in the **arrival**, **stassoc**, **assoc**, **origin**, or **origerr** tables in the IDCX account.

**TUXEDO FILES**

The file required to run *StaPro* in the automatic processing pipeline is `$(CMS_CONFIG)/app_config/distributed/tuxshell/detpro/tuxshell-StaPro.par`. It constructs the *StaPro* command line, which contains the station name, start and end times, and defines the primary *StaPro* parameter file. More information about the Tuxedo files can be found in [IDC6.2.1].

**INITIATING OPERATIONS**

For automatic processing, *StaPro* is automatically initiated when the pipeline is started [IDC5.2.1].

For off-line operation, execute *StaPro* by typing `staPro` and the desired parameter values on the command line.

## VALIDATING INSTALLATION

After *StaPro* is installed, you can run a simple test to verify that *StaPro* is getting the expected parameters from the current environment. This should be done manually and requires specifying the *libpar* LIST STOP option on the end of the command line as follows:

```
$(RELBIN)/StaPro sta=ARCES start-time=1001980800. \  
endtime=1001981400. par=$(AUTOMATIC-DIR)/StaPro/StaPro.par LIST STOP
```

Parameters are listed with the value assigned to them in the parameter file. *StaPro* processing stops at that point and exits. Run this check manually before *StaPro* is used in the automatic processing pipeline.



## References

The following sources supplement or are referenced in document:

- [Gan79] Gane, C., and Sarson, T., *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1979.
- [IDC5.1.1Rev2] Science Applications International Corporation, Veridian Pacific-Sierra Research, *Database Schema, (Part 1, Part 2, and Part 3), Revision 2*, SAIC-00/3057, PSR-00/TN2830, 2000.
- [IDC5.1.3Rev0.1] Science Applications International Corporation, Veridian Pacific-Sierra Research, *Configuration of PIDC Databases*, SAIC-01/3022, PSR-99/TN1114, 2001.
- [IDC5.2.1] Science Applications International Corporation, *IDC Processing of Seismic, Hydroacoustic, and Infrasonic Data*, SAIC-99/3023, 1999.
- [IDC6.2.1] Science Applications International Corporation, *Release 2 Operations and Maintenance—Seismic, Hydroacoustic, and Infrasonic System*, SAIC-00/3000, 2000.
- [IDC6.2.4] Science Applications International Corporation, *Configuration of PIDC Processing Data Files*, SAIC-99/3025, 1999.
- [IDC6.5.2Rev0.1] Science Applications International Corporation, *Distributed Application Control System (DACS) Software User Manual, Revision 0.1*, SAIC-00/3038, 2000.
- [IDC6.5.12] Science Applications International Corporation, *Global Association (GA) Subsystem Software User Manual*, SAIC-01/3003, 2001.

## ▼ References

- [IDC7.1.12] Science Applications International Corporation, *Station Processing (StaPro)*, SAIC-00/3013, 2000.
- [Ril98] Riley, G., Culbert, C., and Donnell, B., *CLIPS Reference Manual*, Volume I, GHG Internet Services, 1998. (online) <http://www.ghgcorp.com/clips/CLIPS.html>
- [Wan97a] Wang, J., Israelsson, H., and North, R. G., *Adaptive Training Approach to Neural Networks for Seismic Phase Identification*, Center for Monitoring Research, CMR-97/29, 1997.
- [Wan97b] Wang, J., *New Default Weights of Neural Networks for Initial Wave-type Identification*, CCB-PRO-97/17, 1997.
- [Wan98c] Wang, J., *Temporary New Default Weights of Neural Networks for Initial Wave-type Identification for Auxiliary 3C Stations at the PIDC*, CCB-PRO-98/13, 1998.



# Glossary

## Symbols

### 3-C

Three-component.

### 2-D

Two-dimensional.

## A

### amplitude

Zero-to-peak height of a waveform in nanometers.

### array

Collection of sensors distributed over a finite area (usually in a cross, triangle, or concentric pattern) and referred to as a single station.

### arrival

Detected signal that has been associated to an event. First, the Global Association (GA) software associates the detection to an event. Later, during interactive processing, many arrivals are confirmed, improved, or added by visual inspection.

## ASCII

American Standard Code for Information Interchange. Standard, unformatted 256-character set of letters and numbers.

### attribute

(1) Database column. (2) Characteristic of an item; specifically, a quantitative measure of a S/H/I detection such as azimuth, slowness, period, or amplitude.

### azimuth

Direction, in degrees clockwise with respect to North, from a station to an event.

## B

### beam

(1) Waveform created from array station elements that are sequentially summed after being steered to the direction of a specified azimuth and slowness. (2) Any derived waveform (for example, a filtered waveform).

## C

### CD-ROM

Compact Disk–Read Only Memory.



**F****f-k**

Frequency versus wavenumber (k) analysis that maps phase power from an array as a function of azimuth and slowness.

**F-statistic**

Measure that indicates the degree of spatial coherence of a waveform across an array of sensors. This measure is approximately equal to the ratio of the spatially coherent energy to the incoherent energy scaled by the number of non-collocated sensors.

**filesystem**

Named structure containing files in sub-directories. For example, UNIX can support many filesystems; each has a unique name and can be attached (or mounted) anywhere in the existing file structure.

**firewall**

Software used to protect a computer or computer network from unauthorized access.

**FTP**

File Transfer Protocol; protocol for transferring files between computers.

**function**

Named section of a program that performs a particular task.

**G****GA**

Global Association.

**GDI**

Generic Database Interface.

**generic object**

Construct used to hold and manipulate data. The type of object determines the data that it can contain. Also known as an object or GObj.

**Global Association**

Subsystem that associates S/H/I phases to events.

**GSETT**

Group of Scientific Experts Technical Test.

**H****hydroacoustic**

Pertaining to sound in the ocean.

**Hz**

Hertz.

**I****IASPEI**

International Association of Seismology and Physics of the Earth's Interior.

**ID**

Identification; identifier.

## ▼ Glossary

**IDC**

International Data Centre.

**IMS**

International Monitoring System.

**infrasonic (infrasound)**

Pertaining to low-frequency (sub-audible) sound in the atmosphere.

**IPC**

Interprocess communication. The messaging system by which applications communicate with each other through *libipc* common library functions. See *tuxshell*.

**K****km**

Kilometer.

**M****magnitude**

Empirical measure of the size of an event (usually made on a log scale).

**Magnitude Description File**

File that maps amptypes and TLtypes to magtypes and specifies magnitude control settings and bulk station correction data.

**magnitude-defining**

See defining magnitude.

**MDF**

See Magnitude Description File.

**N****noise**

Incoherent natural or artificial perturbations of the waveform trace caused by ice, animals migrations, cultural activity, equipment malfunctions or interruption of satellite communication, or ambient background movements.

**O****object**

Same as a generic object.

**ORACLE**

Vendor of the database management system used at the PIDC and IDC.

**P****par**

See parameter.

**parameter**

User-specified token that controls some aspect of an application (for example, database name, threshold value). Most parameters are specified using [*token* = *value*] strings, for example, *dbname=mydata/base@oracle*.

**parameter (par) file**

ASCII file containing values for parameters of a program. Par files are used to replace command line arguments. The files are formatted as a list of [*token* = *value*] strings

**phase**

Arrival that is identified based on its path through the earth.

**PIDC**

Prototype International Data Centre.

**pipeline**

1) Flow of data at the IDC from the receipt of communications to the final automated processed data before analyst review. 2) Sequence of IDC processes controlled by the DACS that either produce a specific product (such as a Standard Event List) or perform a general task (such as station processing).

**primary phase**

First arriving phase recorded at a S/H/I station.

**process**

Function or set of functions in an application that perform a task.

**R****REB**

See Reviewed Event Bulletin.

**Reviewed Event Bulletin**

Bulletin formed of all S/H/I events that have passed analyst inspection and quality assurance review.

**S****s**

Second(s) (time).

**SASC**

Slowness-Azimuth Station Corrections.

**secondary phases**

Phases that arrive after the primary phase.

**seismic**

Pertaining to elastic waves traveling through the earth.

**S/H/I**

Seismic, hydroacoustic, and infrasonic.

**site**

Location of a sensor within a station.

**StaPro**

Station Processing application for S/H/I data.

**station**

Collection of one or more monitoring instruments. Stations can have either one sensor location (for example, BGCA) or a spatially distributed array of sensors (for example, ASAR).

**structure**

Software construct that collects one or more variables, possibly of different types, together under a single name for convenient handling.

## tar

## TLSF

## Transmission Loss Specification File

trigger

- ## Tuxedo

## tuxshell

U

# UNIX

UTC

V

## Velocity Model Specification File

**VMSF**

**W**

**waveform**

Station Processing (StaPro) Software User Manual

# Index

## Symbols

`$(STA)_nnet.log` 9  
`$(STA).par` 8, 42, 59

## Numerics

*3c-azimuth-weight* 23  
*3c-slowness-weight* 23

## A

**amp3c** 7  
     inputs from 17, 21  
*amp3c-table* 17  
**amplitude** 7  
     inputs from 17  
*amplitude-table* 17  
**apma** 7  
     inputs from 17  
*apma-table* 17  
*array-azimuth-weight* 23  
*array-slowness-weight* 23  
**arrival** 7, 48  
     inputs from 17  
*arrival-aux-table* 17  
*arrival-table* 17  
**assoc** 7  
*assoc-table* 18

## B

*bayes-file* 22  
 Bayesian inference 10, 22  
*bayes.tbl* 8, 10, 43

## C

*cfreq1, cfreq2, cfreq3, cfreq4, cfreq5* 20  
*check-attrib-range* 38  
 CLIPS rules 17, 18, 37  
     control facts 39  
     general format 41  
 command line 14  
 common problems 54  
 control parameters 16  
 conventions  
     data flow symbols iv  
     typographical v

## D

database  
     connection parameters 17  
     tables 7  
     use for monitoring *StaPro* 48  
*database* 17  
 data flow symbols iv  
**detection** 7  
     inputs from 17  
*DFX* 2, 11  
*do-loc-mag* 17  
*duration* 16

## ▼ Index

**E**

*end-time* 16  
 error messages  
     interpreting 49  
 error recovery 55  
*ESAL* 4  
*event-confirmation-threshold* 23  
 event location and confirmation  
     parameters 23

**F**

*final-save* 38  
*fpr* 20, 37

**G**

*GA* 2  
*GA\_dist\_depth\_ranges* 8, 10  
*GDI* 5, 9, 54  
*get-largest-s* 38  
*group-amplitude-tolerance* 22  
*group-delaz-factor* 22  
*group-first-s-p-time* 22  
*group-s-p-time-factor* 22

**H**

hardware requirements 11  
*hf-band-high-cut* 24  
*hf-band-low-cut* 24  
*h-phase* 25  
 H phases 23  
*hydro\_\$(STA).weights* 8  
**hydro\_features** 7, 31  
     inputs from 17  
*hydro\_sta.wts* 43  
 hydroacoustic parameters 23  
*hydro-look-back-time* 24

*hydro-nnet-log-file* 24, 26  
*hydro-nnet-weights-file* 24, 26  
*hydro-table* 17

**I**

IDCX 59  
*IMSPAR* 14  
*infra-delaz-factor* 26  
*infra-i-phase* 26  
*infra-look-back-time* 25  
*infra-lower-velocity* 26  
*infra-max-grouping-time* 26  
*infra-n-phase* 26  
 infrasonic parameters 25  
*infra-unknown-i-phase* 26  
*infra-upper-velocity* 26  
*init-phase-nnet* 37  
 installation procedures 57  
*intermediate-p-phase* 22  
*intermediate-p-vel-boundary* 22  
*ipnnwts.tbl* 8, 10, 43  
*ipr* 20, 37  
*iwt-by-velocity* 38

**L****lastid**

    initialization 60  
*lf-band-high-cut* 24  
*lf-band-low-cut* 24  
*lg-max-velocity* 22  
*lg-min-velocity* 22  
*lg-tol* 22  
*libloc* 23  
*libpar* 61  
*local-delaz-factor* 21  
*local-max-s-p-time* 21  
*local-min-p-p-separation* 21  
*local-s-phase* 21  
*locator-outfile-name* 23



*locator-verbose* 23  
 log files  
     maintaining 45  
     monitoring 49

## M

*mag-def-file* 23  
*magtype* 23  
*max-delslo* 21  
*max-group-amplitude-factor* 22  
*max-grouping-time-no-azimuth* 22  
*max-noise-velocity* 21  
*max-p-coda-arrival-time* 22  
*max-p-hvrat* 21  
*maxrec* 18  
*max-rg-time* 21  
*max-teleseismic-amplitude-factor* 21  
 MDF 8, 10, 59  
*min-grouping-amplitude-factor* 22  
*min-lg-frequency* 21  
*min-p-freq* 21  
*min-p-rect* 21  
*min-sn-p-time* 22  
*min-teleseismic-amplitude-factor* 21  
*min-teleseismic-velocity* 21

## N

*n-duration-lim* 25  
 neural network 18  
     hydroacoustic inputs 31  
     hydroacoustic weights file format 35  
     seismic input 27  
     seismic weights file format 30  
     training 6  
     using 26  
     weights 44  
     weights files 10  
*nnet\_train* 6  
*nnet-log-file* 26

*nnet-weights-file* 26  
*n-num-cross-lim* 25  
*noise-fkqual* 21  
*noise-fkqual-fstat* 21  
*noise-phase* 22  
*noise-screener* 38  
*n-phase* 25  
 N phases 23  
*n-ratio-lim* 25  
*n-time-spread-lim* 25  
*n-total-time-lim* 25

## O

obtaining software 58  
 operational procedures 13  
*o-phase* 25  
**origerr** 7  
*origerr-table* 18  
**origin** 8  
*origin-table* 18

## P

parameters  
     control 16  
     database connection 17  
     event location and confirmation 23  
     hydroacoustic 23  
     infrasonic 25  
     phase identification 22  
     setting 15  
     signal grouping 21  
     signal-type determination 18  
 passwords 45  
*pg-tol* 22  
*phase-distance-file* 22  
 phase-identification parameters 22  
*primary-time-weight* 23  
*process.par* 9, 14, 15, 45  
     location 59

## ▼ Index

*p-s-velocity-threshold* 21

**R**

*regional-min-s-p-time* 22

requirements

software and hardware 11

Retrieve Subsystem 2

*rg-max-velocity* 22

*rg-min-velocity* 22

*rg-tol* 22

**S**

SASC 6, 10, 59

*sasc.\$(STA)* 42

*sasc-dir-prefix* 17

*save* 37

*secondary-time-weight* 23

security 45

seismic weights file 28

*semi-major-axis-threshold* 23

*shared.par* 9, 15, 42, 43

signal-grouping parameters 21

signal-type determination parameters 18

**site** 8, 42, 43

*site-table* 17

*sn-confidence-threshold* 22

software requirements 11

*s-split-parm* 22

*sta* 16

*standard-beam-chan* 16

*StaPro*

functions 5

functions bound to *CLIPS* 37

maintenance 44

monitoring 48

overview 2

performance 6

*StaPro\_loc.err* 9

*StaPro.par* 8, 9, 15, 26

location 59

*start-time* 16

startup 14

**stassoc** 8

*stassoc-table* 18

station

adding new 42

tuning 42

*statype* 16, 43

**T**

*teleseismic-amplitude-tolerance* 21

*teleseismic-delaz-factor* 21

*teleseismic-max-p-p-time* 21

*teleseismic-phase* 21

*tis\_server* 11

TLSF 10, 42, 59

*tl-spec-file* 23

*t-num-cross-lim* 25

*t-phase* 25

T phases 23

*t-ratio-lim* 25

troubleshooting 47

*t-time-spread-lim* 25

*tt-phases* 23

tuning stations 42

Tuxedo queues 11

*tuxshell* 2, 9, 11

*tuxshell-StaPro.par* 9, 14, 60

typographical conventions v

**U**

*unknown-h-phase* 25

*unknown-o-phase* 25

*unknown-primary-phase* 22

*unknown-secondary-phase* 22

*unknown-teleseismic-phase* 22

*unknown-t-phase* 25

## V

*vendor* 18  
*verbose* 17  
*verbose-check* 38  
*vmodel-spec-file* 23  
VMSF 8, 10, 42

## W

*WaveExpert* 2  
*WorkFlow* 48

